MITRE

Systems Engineering Guide

Collected wisdom from MITRE's
systems engineering experts

Systems Engineering Guide

MITRE

Produced by MITRE Corporate Communications and Public Affairs
International Standard Book Number: 978-0-615-97442-2
Printed in the United States of America on acid-free paper

The views, opinions, and/or findings contained in this book are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation. This book discusses observations and ideas, and The MITRE Corporation expressly disclaims any warranty of any kind. Although products are discussed in this book, nothing in this book should be construed as an endorsement of any kind. The trademarks used herein belong to their respective holders.

Approved for public release; distribution unlimited. Case Numbers 10-4072 and 12-1089.

The MITRE Corporation
202 Burlington Road • Bedford, MA 01730-1420 • (781) 271-2000
7515 Colshire Drive • McLean, VA 22102-7539 • (703) 983-6000
www.mitre.org
segteam@mitre.org

Acknowledgments

The MITRE Systems Engineering Guide (SEG) was first launched in March 2010 as an internal MITRE resource. In late 2010, a government-only version was rolled out in response to many requests from MITRE staff to use it as a shared resource with their customers. In June 2011, an HTML version was published on www.mitre.org as a contribution to the wider systems engineering community. The rollout of the public SEG resulted in requests for it to be made available for popular mobile platforms, and in September 2012 an eBook version was posted on www.mitre.org in formats for the iPad, iPhone, Android, Kindle, and compatible devices. The SEG has been visited hundreds of thousands of times by individuals across the world. Now it is available in hardcopy form.

The SEG is the result of an effort involving nearly 200 individuals from across MITRE.

The seminal idea for capturing the corporation's accumulated wisdom on a wide variety of important and timely systems engineering topics in a single, central location came from MITRE Corporate Chief Engineer Dr. Louis S. Metzger. He inspired the vision of the SEG as a resource that provides an "in the trenches" view of the typical problems, pitfalls, conundrums, and tight corners that practicing systems engineers are likely to find themselves in, together with best practices and lessons learned to avoid or mitigate the problems and enhance the likelihood of success. In this way, the SEG complements the many excellent systems engineering resources currently available.

An undertaking this ambitious requires a core team to organize and orchestrate the myriad details—large and small—that are essential to overall success. Special thanks are due to SEG core team members: Robin A. Cormier, Spurgeon T. Norman, Jr., Deborah L. Schuh, Peter A. Smyton, Dr. Robert S. Swarz, and Frederick C. Wendt.

The vast majority of contributors authored individual articles and are among the most senior and respected MITRE technical staff. The members of the core team stand on the shoulders of these experts.

George Rebovich, Jr.
Team Leader

SE Guide Contents

Acquisition Systems Engineering 483

Introduction

Welcome to the MITRE Systems Engineering Guide (SEG). The primary purpose of the SEG is to convey The MITRE Corporation's accumulated wisdom on a wide range of systems engineering subjects—sufficient for understanding the essentials of the discipline and for translating this wisdom into practice in your own work environment.

The MITRE Systems Engineering Guide (SEG) has more than 600 pages of content and covers more than 100 subjects. It has been developed *by* MITRE systems engineers *for* MITRE systems engineers. Systems engineering is a team sport, so although the SEG is written "to" a MITRE systems engineer, most of the best practices and lessons learned are applicable to all members of a government acquisition program team, whatever their particular role or specialty.

This introduction provides guidance on how to navigate the pages of the SEG and benefit from doing so. It covers the practical matters—the organization, use, and roots of the SEG, what you should (and should *not*) expect from its articles, and how you can access and respond to the latest SEG information on MITRE's website.

How the SEG Is Organized

Setting the Context for the Systems Engineering Guide

- **The Evolution of Systems Engineering—**provides a working definition of the discipline and traces its evolutionary arc into the future.
- **The Essence of MITRE Systems Engineering—**introduces how our sponsors perceive MITRE systems engineering roles and responsibilities, and how we at MITRE interpret those expectations.
- **The Systems Engineering Guide—**Three "meaty" sections partitioned into topics and articles:
  - **Enterprise Engineering**—explains how to take a comprehensive view of systems engineering activities at different scales of the customer enterprise, offers techniques for engineering information-intensive enterprises that balance local and global needs, and covers how to provide systems engineering support to governance activities.
  - **Systems Engineering Life-Cycle Building Blocks**—is organized around the fundamentals of setting up engineering systems regardless of the specific life-cycle methodology used by the supporting sponsor or customer.
  - **Acquisition Systems Engineering**—is centered on how MITRE systems engineering fits into and supports government acquisition programs.

How to Use the Systems Engineering Guide (SEG)

The first time you access the SEG, read the two expository pieces—The Evolution of Systems Engineering and The Essence of MITRE Systems Engineering—in their entirety. They are intended to set the context for the material in the three major sections.

Then, take some time to familiarize yourself with the SEG by reading the section-level introductions and sampling a topic or two and a few articles.

To support your work program or SE educational activities, come back to specific topics and articles in the SEG as needed.

Systems Engineering Competency Model

The SEG organization and perspective were inspired by and based on the MITRE Systems Engineering Competency Model (SECM). MITRE uses the SECM primarily for competency assessments (self and manager) and development activities, including an internal systems engineering curriculum. The competency model is included with the SEG on www.mitre.org.

Each article in the SEG contains a brief *MITRE Systems Engineering Roles & Expectations* statement distilled from the competency model. Although we believe that much in the SEG and SECM is applicable to others, the articles should be used as references to be tailored to your specific objectives and circumstances.

What You Will Find in an Article

The articles are written as if the author is speaking directly to a MITRE technical staff member involved in an FFRDC-related systems engineering activity on a government program or to someone who wants to learn more about a particular systems engineering perspective. The authors are MITRE systems engineering practitioners with substantial experience in a particular subject area.

Each article attempts to convey where MITRE systems engineering typically fits in the big picture of government participants and commercial contractors and clarifies how MITRE's role differs from that of the other players. Each article follows the same basic construct:

- The authors were asked, "What are the common problems, pitfalls, conundrums, and tight corners that MITRE systems engineers are likely to find themselves in when working in this subject area?"
- For each problem or conundrum, the authors answered the question, "What wisdom is there to convey to avoid or mitigate problems or enhance the likelihood of success?"
- The wisdom is conveyed in a set of succinct best practices and lessons learned.
- When an important conundrum is identified, when possible, potential approaches are suggested for solving the problem.

- Each article cites references and resources for additional reading. Be sure to check them out if you are interested in more details.

What the SEG Is *Not*

The SEG is not intended to provide guidance on every possible issue under the "systems engineering sun." A complete discussion on even one topic could probably fill volumes. Nor is it intended to serve as a compendium of *Systems Engineering 101* tutorials. A rich set of resources, within MITRE and beyond, can be tapped into for educational purposes. And though the SEG is based on the collective experience of MITRE systems engineers across the company, it is not intended to serve as a resource on detailed sponsor- or customer-specific systems engineering policies, practices, or processes.

Systems engineering is a dynamic and evolving discipline, and we are actively evolving the SEG to keep pace with that change. Be sure to visit MITRE's online version of the SEG periodically at www.mitre.org to see what's new.

Finally, we hope that you find this material of interest. If you have comments or feedback, please contact us at segteam@mitre.org.

Setting the Context for the Systems Engineering Guide

The next section presents two expository pieces—*The Evolution of Systems Engineering* and *The Essence of MITRE Systems Engineering*—detailing, respectively, how systems engineering has evolved as a discipline and how MITRE's systems engineering practice is shaped by our role as an operator of Federally Funded Research and Development Centers (FFRDCs). Together, these pieces are intended to set the context for your use of MITRE's Systems Engineering Guide (SEG).

The Evolution of Systems Engineering

The twenty-first century is an exciting time for the field of systems engineering. Advances in our understanding of the traditional discipline are expanding. At the same time, new forms of systems engineering are emerging to address the engineering challenges of systems-of-systems (SoS) and enterprise systems. Even at this point in their evolution, these new forms are evincing their own principles, processes, and practices. Some are different in degree than engineering at the system level, whereas others are different in *kind*.

Although it is impossible to predict how the traditional and new forms of systems engineering will evolve, it is clear even now that a long and robust future lies ahead for all. Increases in technology complexity have led to new challenges in architecture, networks, hardware and software engineering, and human systems integration. At the same time, the scale at which systems are engineered is exceeding levels that could have been imagined only a short time ago. As a consequence, all forms of systems engineering will be needed to solve the engineering problems of the future, sometimes separately but increasingly in combination.

What Is Systems Engineering?

The term *systems engineering* can be traced back at least to the 1940s, but to this day no single, universal definition of the term exists. Frequently, *systems engineering* is defined by the context in which it is embedded. One definition of the classical practice of systems engineering is, "an interdisciplinary approach to translating users' needs into the definition of a system, its architecture and design through an iterative process that results in an effective operational system. Systems engineering applies over the entire life cycle, from concept development to final disposal [1]."

Systems Engineering Life Cycle

Systems engineering models and processes are usually organized around the concept of a *life cycle*. Like the definition of systems engineering, the detailed conceptualization of life cycle is by no means unique across the communities that employ the discipline.

The International Council on Systems Engineering (INCOSE) systems engineering process is a widely recognized representation of classical systems engineering [2]. ISO/IEC 15288 [3] is an international systems engineering standard covering processes and life-cycle stages. It defines a set of processes divided into four categories: technical, project, agreement, and enterprise. Sample life-cycle stages include concept, development, production, utilization, support, and retirement. The U.S. Department of Defense uses the following phases: materiel solution analysis, technology development, engineering and manufacturing development, production and deployment, and operations and support [4].

Although the detailed views, implementations, and terminology used to articulate the systems engineering life cycle differ across MITRE's sponsors and customers, they all share fundamental elements, depicted in Figure 1 in the V-model [5]. This is a common graphical representation of the systems engineering life cycle. The left side of the V represents concept development and the decomposition of requirements into functions and physical entities that can be architected, designed, and developed. The right side of the V represents integration of these entities (including appropriate testing to verify that they satisfy the requirements) and their ultimate transition into the field, where they are operated and maintained.

The model of systems engineering used in this guide is based on the "V" representation. Note, however, that the system life cycle is rarely, if ever, as linear as this simplified discussion might imply. There are often iterative cycles, skipped phases, overlapping elements, etc. Additionally, important processes and activities apply to more than one phase in a system life cycle, which are better envisioned as threading through or overarching the other building

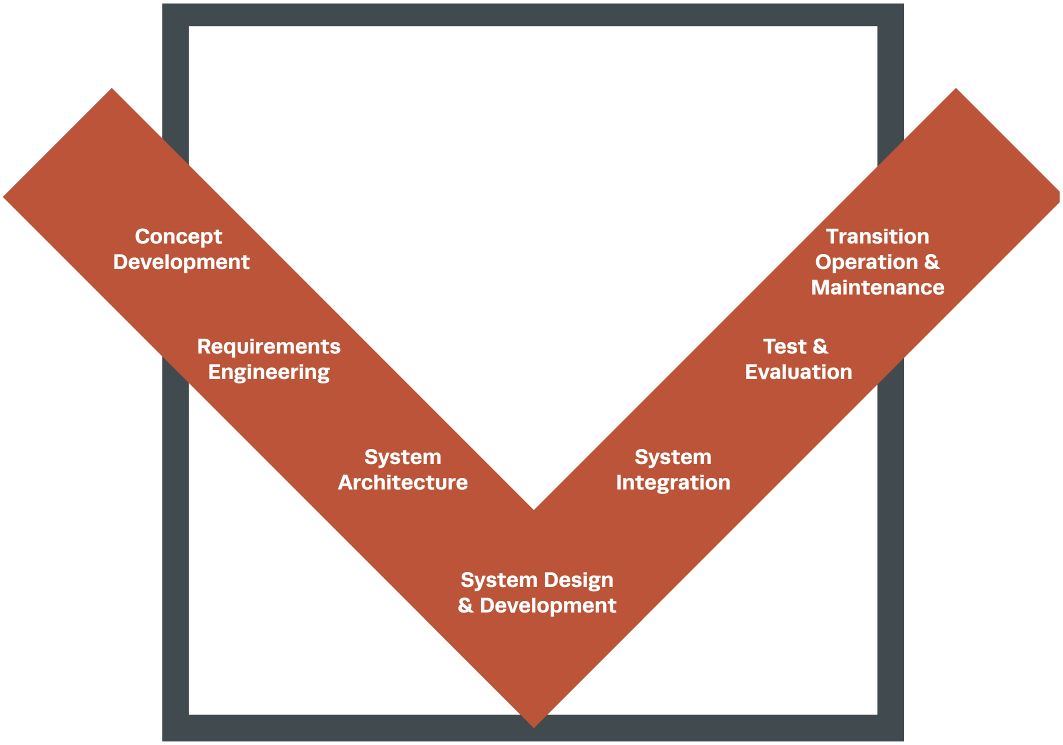


Figure 1. V–Model of Systems Engineering Life Cycle

blocks. Risk identification and management is one example. Consult the SE Life-Cycle Building Blocks and Acquisition Systems Engineering sections for details.

Numerous variations on the classical systems engineering life cycle can be found, including incremental or spiral developments that mitigate uncertainties in long-range requirements or funding of the system under development as well as evolutionary approaches for navigating uncertainties in enabling technology maturity. All three sections of the guide—Enterprise Engineering section, SE Life-Cycle Building Blocks section, and Acquisition Systems Engineering section—contain discussions on these variations.

Conditions for Effective Systems Engineering

As already noted, systems engineering is normally defined and shaped by the context or environment in which it is embedded. The classical systems engineering approach is tailored to and works best in situations in which all relevant systems engineering factors are largely under the control of or can at least be well understood and accommodated by the systems engineering organization or the program manager. In general terms, this is when system requirements are relatively well established, technologies are mature, the system is being developed for a single or relatively homogeneous user community, and a single individual has management and funding authority over the program. Even then, these conditions, while necessary, are rarely sufficient to ensure success. What is needed, however, are a strong government program office capable of a peer relationship with the contractor; effective architecting, including problem definition, evaluation of alternative solutions, and analysis of execution feasibility; careful attention to program management and systems engineering foundational elements; selection of an experienced, capable contractor; and effective performance-based contracting.

A Changing Landscape—Systems of Systems

With the increased emphasis on capabilities and networking, MITRE's sponsors and customers are recognizing the criticality of effective end-to-end performance of SoS to meet operational user needs. Though most government acquisition policies and processes continue to focus on the development and evolution of individual systems, their requirements are increasingly based on assessments of gaps in user capabilities that require integration across individual systems to be enabled. Increasingly, the role of systems engineering is turning to the engineering of SoS to provide these capabilities.

One working definition of SoS is "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities [6]." Both individual systems and SoS are considered systems because each consists of parts, relationships, and a "whole" that is greater than the sum of the parts. However, not all systems are SoS. Rather, SoS systems engineering deals with "[the] planning, analyzing, organizing and integrating of the capabilities of a mix of existing and new development systems into an SoS capability greater than the sum of the capabilities of the constituent parts [6]." SoS may deliver capabilities by combining multiple collaborative, autonomous-yet-interacting systems. The mix of systems may include existing, partially developed, and yet-to-be-designed independent systems.

SoS can take different forms, as shown in Table 1 [7, 8]. The Global Information Grid is an example of a virtual SoS. Communities of interest are examples of a collaborative SoS. The Missile Defense Agency Ballistic Missile Defense System is an example of an acknowledged SoS, and the U.S. Army Future Combat System is an example of a directed SoS.

Increasingly, MITRE sponsors and customers are facing the challenges of acknowledged SoS, defined in Table 1. This calls for capability management and SE at the SoS level while maintaining the management and technical autonomy of systems contributing to the SoS capability objectives.

Table 1. Types of Systems of Systems

Type	Definition
Virtual	Virtual SoS lack a central management authority and a centrally agreed-on purpose for the system of systems. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely on relatively invisible mechanisms to maintain it.
Collaborative	In collaborative SoS, the component systems interact more or less voluntarily to fulfill agreed-on central purposes. The Internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.
Acknowledged	Acknowledged SoS have recognized objectives, a designated manager, and resources. However, the constituent systems retain their independent ownership, objectives, funding, development, and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system.
Directed	Directed SoS are those in which the integrated system of systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might want to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.

A typical strategy for providing end-to-end support for new capability needs is to add functionality to assets already in the inventory. In most cases, these systems continue to be used for their original requirements. Consequently the ownership or management of these systems remains unchanged, and they continue to evolve based on their own development and requirements processes and independent funding.

The resulting dual levels of management, objectives, and funding create management challenges for both the SoS and the systems, especially when their objectives are not well aligned. In turn, these management challenges pose technical challenges for systems engineers, especially those working on the SoS. Table 2 summarizes differences between systems and acknowledged SoS that have particular implications for engineering SoS.

The differences summarized in Table 2 lead to differences in SoS engineering. Some are differences in degree, and others are differences in kind. These are briefly outlined here, and the references provide a more detailed discussion.

- SoS systems engineers must be able to function in an environment where the SoS manager does not control all of the systems that impact the SoS capabilities and where the stakeholders have interests beyond the SoS objectives [9, pp. 11–12].
- SoS SE must balance SoS needs with individual system needs [9, p. 12].
- SoS SE planning and implementation must consider and leverage development plans of the individual systems [9, pp. 13–14].
- SoS SE must address the end-to-end behavior of the ensemble of systems, addressing key issues affecting the behavior [9, pp. 14–15].

The discipline of SoS systems engineering is still in its infancy. Nevertheless, a set of SoS systems engineering principles is beginning to emerge from a U.S. Department of Defense (DoD) initiative to understand and differentiate engineering of these complex, increasingly common entities from individual systems [10]. These guiding principles are briefly noted here and are discussed in more detail in the references.

- Address organizational as well as technical issues when making SE trades and decisions [9, p. 21].
- Acknowledge the different roles of systems engineers at the system vs. the SoS level and the relationship between the different SE approaches taken at each of the levels [9, pp. 21–22].
- Conduct balanced technical management of the SoS [9, p. 22].
- Use an architecture based on open systems and loose coupling [9, p. 23].
- Focus on the design strategy and trade-offs when the formal SoS is first established and throughout the SoS evolution [9, p. 23].

Table 2. Comparison of Systems and Systems of Systems [9, p. 13]

Aspect of Environment	System	Acknowledged System of Systems
Management & Oversight		
Stakeholder Involvement	Clearer set of stakeholders	Stakeholders at both system level and SoS levels, including system owners with competing interests and priorities. In some cases, the system stakeholder has no vested interest in the SoS; all stakeholders may not be recognized.
Governance	Aligned program management and funding	Added levels of complexity due to management and funding for both the SoS and individual systems. SoS does not have authority over all of the systems.
Operational Environment		
Operational Focus	Designed and developed to meet operational objectives	Called on to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives.
Implementation		
Acquisition	Aligned with acquisition milestones, documented requirements, program has a systems engineering plan	Added complexity due to multiple system life cycles across acquisition programs, involving legacy systems, developmental systems, new developments, and technology insertion. Typically they have stated capability objectives upfront which may need to be translated into formal requirements.
Test & Evaluation	Test and evaluation of the system is generally possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems' life cycles, given the complexity of all the moving parts and potential for unintended consequences.
Engineering & Design Considerations		
Boundaries & Interfaces	Focuses on boundaries and interfaces for the single system	Focus is on identifying systems that contribute to the SoS objectives and enabling the flow of data, control, and functionality across the SoS while balancing needs of the systems.
Performance & Behavior	Performance of the system to meet specified objectives	Performance across the SoS satisfies SoS user capability needs while balancing needs of the systems.

Engineering the Enterprise [11, 12]

MITRE's sponsors, customers, and the users of the operational systems we help engineer are in the midst of a major transformation driven by and deriving largely from advances in information technology.

The rate of technical change in information processing, storage, and communications bandwidth is enormous. Expansions in other technologies (e.g., netted sensors) have been stimulated and shaped by these changes. The information revolution is reducing obstacles to interactions among people, businesses, organizations, nations, and processes that were previously separated in distance or time. Somewhat paradoxically, future events in this information abundant world are harder to predict and control, with the result that our world and our role as systems engineers are becoming increasing complex.

This new complexity is a consequence of the interdependencies that arise when large numbers of systems are networked together to achieve some collaborative advantage. It is further intensified by rapid technology changes. When networked systems are each individually adapting to both technology and mission changes, then the environment for any given system or individual becomes essentially unpredictable. The combination of large-scale interdependencies and unpredictability creates an environment that is fundamentally different from that at the system or SoS level.

Examples in which this new complexity is evident include the U.S. Federal Aviation Administration's National Airspace System, the DoD's Global Information Grid, the Internal Revenue Service's Tax Systems, and the Department of Homeland Security's Secure Border Initiative's SBInet.

As a result, systems engineering success expands to include not only that of an individual system or SoS, but of the network of constantly changing systems as well. To successfully bring value to these enterprise system users requires the disciplined methods and "big picture" mindset of the classical forms of systems engineering, plus new methods and mindsets aimed at addressing the increased complexity.

Because our customers' needs are driving the trend toward collaborative advantage and adaptability, we must evolve our methods to these changing conditions. This situation is characterized by several specific characteristics:

- Our customers face extremely complex problems in which stakeholders often disagree on the nature of the problems as well as the solutions (i.e., technical and social).
- Their missions are changing rapidly and unpredictably—thus systems must interoperate in ways that their original developers never envisioned.
- Even without a predefined direction, the systems will keep evolving and responding to changing needs and emerging opportunities—the network is inherently adaptive.

- People are integral parts of the network, and their purposeful behavior will change the nature of the network—individual systems must be robust to changes in their environment.

Thus the systems that we help engineer are facing additional, fundamentally different challenges. Nevertheless, when a system is bounded with relatively static, well-understood requirements, the classical methods of systems engineering are applicable and powerful. It is the increased complexity of problems and solutions that has caused us to extend the systems engineering discipline into a domain we call *enterprise systems engineering*.

What do we mean by an enterprise? Enterprise refers to a network of interdependent people, processes, and supporting technology not fully under the control of any single entity. In business literature, an enterprise frequently refers to an organization, such as a firm or government agency, and in the computer industry, it refers to any large organization that uses computers. The MITRE definition emphasizes the interdependency of individual systems and even systems of systems. We include firms, government agencies, large information-enabled organizations, and any network of entities coming together to collectively accomplish explicit or implicit goals. This includes the integration of previously separate units. The enterprise displays new behaviors that emerge from the interaction of the parts. Examples of enterprises include:

- A military command and control enterprise of organizations and individuals that develop, field, and operate command and control systems, including the acquisition community and operational organizations and individuals that employ the systems.
- A chain hotel in which independent hotel properties operate as agents of the hotel enterprise in providing lodging and related services, while the company provides business service infrastructure (e.g., reservation system), branding, etc.

What do we mean by enterprise systems engineering? This domain of systems engineering concentrates on managing uncertainty and interdependence in an enterprise. It encompasses and balances technical and non-technical aspects of the problem and the solution. It fits within the broad, multidisciplinary approach of systems engineering and is directed toward building effective and efficient networks of individual systems to meet the objectives of the whole enterprise.

In performing enterprise systems engineering, we engineer the enterprise and we engineer the systems that enable the enterprise. In particular, we help customers shape their enterprises, aligning technology to support goals. We support their business planning, policy-making, and investment strategies. We also determine how the individual systems in the enterprise perform and how they affect each other.

At MITRE, we consider enterprise systems engineering as a domain that focuses on complexity in the broader practice of systems engineering. It is not a replacement for classical methods, and often both classical systems engineering and enterprise systems engineering approaches must be applied in combination to achieve success.

We are learning and evolving enterprise systems engineering as we are doing it. Several basic tenets in the practice are apparent even at this early stage of its evolution:

- **Systems thinking:** Seeing wholes, interrelationships, and patterns of change.
- **Context awareness:** Being mindful of the political, operational, economic, and technical influences and constraints.
- **Accepting uncertainty:** Acknowledging that some problems cannot be solved by prescriptive or closed-form methods.
- **Complex systems evolution:** Drawing from the fundamental principles in the sciences of evolution, ecology and adaptation (e.g., considering variety, self-organization, and selection).
- **Matching practice to the problem:** Knowing when and under what circumstances to apply prescriptive methods and when to apply complex systems principles and associated practices.

The SEG's Enterprise Engineering section and the references provided in the articles are the primary source for enterprise systems engineering subjects. This is a rapidly changing domain of systems engineering.

References and Resources

1. Committee on Pre-Milestone A Systems Engineering, 2009, *Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Acquisition*, The National Academies Press.
2. International Council on Systems Engineering (INCOSE), INCOSE Systems Engineering Handbook.
3. ISO/IEC 15288, 2002, Systems Engineering—System Life Cycle Processes.
4. Department of Defense Instruction Number 5000.02, December 8, 2008, Operation of the Defense Acquisition System.
5. Wikipedia contributors, "V-Model," Wikipedia (accessed January 13, 2010).
6. Department of Defense, October 14, 2004, "System of Systems Engineering," Defense Acquisition Guidebook, Washington, DC.
7. Maier, M., 1998, "Architecting Principles for Systems-of-Systems," *Systems Engineering*, Vol. 1, No. 4, pp 267–284.

8. Dahmann, J., and K. Baldwin, April 7–10, 2008, "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," IEEE Systems Conference, Montreal, Canada.

9. Office of the Undersecretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), August 2008, Systems Engineering Guide for Systems of Systems, Washington, DC.

10. Baldwin, K., June 28, 2007, "Systems of Systems: Challenges for Systems Engineering," INCOSE SoS SE Panel.

11. The MITRE Corporation, August 2007, Evolving Systems Engineering, Bedford, MA.

12. Rebovich, G., Jr., April 2006, "Systems Thinking for the Enterprise: New and Emerging Perspectives," *Proceedings of the 2006 IEEE International Conference on Systems of Systems*.

The Essence of MITRE Systems Engineering

The previous section, The Evolution of Systems Engineering, notes that the systems engineering discipline is defined by the context or environment in which it is embedded. This companion section describes more specifically how the distinctive attributes of MITRE systems engineering are shaped by the expectations of our sponsors and customers and further formed by our corporate interpretation of the quality systems engineering required to meet those expectations.

Sponsor Expectations for MITRE Systems Engineering

The U.S. Federal Acquisition Regulation (FAR) part 35.017 sets forth federal policy on the establishment and use of Federally Funded Research and Development Centers (FFRDCs) and related sponsoring agreements [1]. A portion is excerpted below.

> 35.017 Federally Funded Research and Development Centers.
>
> (a) *Policy.*
>
> …
>
> …
>
> (2) An FFRDC meets some special long-term research or development need which cannot be met as effectively by existing in-house or contractor resources. FFRDC’s enable agencies to use private sector resources to accomplish tasks that are integral to the mission and operation of the sponsoring agency. An FFRDC, in order to discharge its responsibilities to the sponsoring agency, has access, beyond that which is common to the normal contractual relationship, to Government and supplier data, including sensitive and proprietary data, and to employees and installations equipment and real property. The FFRDC is required to conduct its business in a manner befitting its special relationship with the Government, to operate in the public interest with objectivity and independence, to be free from organizational conflicts of interest, and to have full disclosure of its affairs to the sponsoring agency. It is not the Government’s intent that an FFRDC use its privileged information or access to installations equipment and real property to compete with the private sector.
>
> …
>
> …
>
> (4) Long-term relationships between the Government and FFRDC’s are encouraged in order to provide the continuity that will attract high-quality personnel to the FFRDC. This relationship should be of a type to encourage the FFRDC to maintain currency in

its field(s) of expertise, maintain its objectivity and independence, preserve its familiarity with the needs of its sponsor(s), and provide a quick response capability.

Some phrases from this excerpt stand out as particularly important factors that influence the way in which MITRE executes its systems engineering roles and responsibilities:

- Meets some special long-term research or development need which cannot be met [otherwise]
- Private sector resources
- Access, beyond that which is common to the normal contractual relationship
- Operate in the public interest with objectivity and independence
- Free from organizational conflicts of interest
- Full disclosure of its affairs to the sponsoring agency
- Not…compete with the private sector
- Currency in its field(s) of expertise
- Familiarity with the needs of its sponsor(s)

MITRE's individual FFRDC sponsoring agreements further shape how we perceive and practice systems engineering [2, 3, 4, 5]. The FFRDC sponsoring agreements for the NSEC [National Security Engineering Center], CAASD [Center for Advanced Aviation System Development], CEM [Center for Enterprise Modernization], and SEDI [Homeland Security Systems Engineering and Development Institute] further delineate the purpose and role of each FFRDC, its core work, relationship to the sponsoring organization, and other details of its operation. Despite obvious differences among the sponsoring agreements, two consistent themes are evident: Each FFRDC is expected to be doing appropriate work that answers the nation's needs, and that work needs to be done well. Within MITRE, we sometimes use the shorthand "do the right work" when referring to the former and "do the work right" when referring to the latter. These two fundamental characteristics of quality systems engineering are understood and practiced by MITRE. The following excerpts from each of the four sponsoring agreements illustrate these aspects of MITRE systems engineering.

Do the Right Work

- The work performed…will…be…of both long-term and immediate homeland security concern…
- Identification of critical capability gap[s]…particularly in areas where technology… contribute[s] substantially to solutions.
- Subjects integral to the mission and operations of the sponsoring offices.
- Provid[e] technical and integration expertise…particularly in the evolution of the most complex and critical homeland security programs.

- Promote compatibilities across the various homeland security platforms and equipment…through…improved interoperability and information sharing within the homeland security enterprise.
- Work on the most complex homeland security systems that will evolve capabilities…
- Help the Department develop a DHS system of systems approach…
- Address the long- and short-term evolutionary change necessary to modernize the NAS.
- Development and evaluation of plans for the evolution and integration of ATM system capabilities.
- Problems that do not stand alone but are so linked to others that highly specific analysis may be misleading.
- Issues that cannot be formulated sharply enough in advance.
- Unprecedented problems that require unique research methods.
- Perform studies, analysis and concept formulation for continued…modernization and development of the NAS.
- Works with DoD [Department of Defense] to research, develop, integrate, field, sustain and modernize timely, affordable and interoperable C4ISR [Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance] solutions, systems and technology.
- Provid[e] enterprise systems engineering and integration support throughout the C4ISR mission area.]
- Help identify, define, and recommend solutions to problems as a trusted partner of the Sponsors' management team.
- Focus…on core work that promotes C4ISR integration/interoperability.
- [Maintains] an end-to-end understanding of the C4ISR mission area with emphasis on enterprise architectures that enable increasingly advanced and more fully integrated systems of systems, system acquisition (including technical support to source selection), integration of commercial and military technologies and interoperability.

Do the Work Right

- Produces high-quality work of value to the sponsors
- …performance of objective, high-quality work…
- Provide the government with the necessary expertise to provide best lifecycle value…
- Develop and promote standardization of effective and efficient system engineering best practices…
- The work performed…will…be authoritative…
- …purpose is to provide special technical expertise

- Simultaneously direct…efforts to the support of individual programs and projects for enterprise modernization, assuring that these individual programs and projects operate effectively with one another and efficiently support the business objectives of the Government.
- Provide exceptional technical competence in support of the Sponsors' design and pursuit of mission goals.
- Partner with the Sponsors in pursuit of excellence in public service.
- Maintain a commitment to technical excellence…in everything it does.
- Promotion of technical excellence…will be paramount.
- …shall be responsible to the FAA with regard to the progress and quality of…NAS development efforts undertaken by it.
- …staff…encouraged to publish…in professional journals…to have the quality of such work subject to peer scrutiny.
- maintaining objectivity and high technical quality.
- …maximize value…
- …while serving the immediate needs of the many individual programs it supports, the C3I FFRDC aligns its work program to assist in achieving integrated enterprise capabilities…
- …information as an enterprise asset to be shared…

MITRE Expectations for Quality in Systems Engineering [6]

Quality in MITRE's systems engineering includes aspects of both delivering an inherently good product or service and meeting external expectations. For MITRE, external expectations are set by multiple stakeholders, including not only our immediate customers but also the end users of the capabilities we help create, our FFRDC sponsors (and those above them who set expectations for FFRDCs more generally), and our Board of Trustees (who are external to day-to-day company affairs). For the most part, the higher level expectations from our sponsors and Board align with each other and with our internal aspirations for "good" as embodied by our strategic framework. They also align with how MITRE can and should uniquely contribute to meeting end user needs. These alignment points include:

1. Working in the public interest on issues of critical national importance by…
2. Proactively applying systems engineering and advanced technology to bring…
3. Timely and innovative/creative solutions to key, hard problems, balancing…
4. Technical feasibility with economic and political practicality, and leveraging…
5. Breadth and depth of engineering with mission/business domain knowledge, while…
6. Providing an integrating perspective across boundaries, and always…
7. Retaining objectivity and being cost effective in our work.

To meet these expectations we need to be doing appropriate work that answers the nation's needs, and we need to do it well. This is the key requirement that cuts across our four sponsoring agreements. We also need to satisfy our immediate customers. And we need to invest in developing quality relationships with decision makers, stakeholders, and our customers, to shape our work and present results so that they have the impact they deserve. Meeting our customers' expectations requires that we provide value in the quality of our contributions.

Therefore, quality in MITRE systems engineering can be defined as the degree to which the results of systems engineering meet:

1. The higher level expectations for our FFRDCs—resulting in usability and value for end recipients.
2. Expectations of our immediate customers—service and performance.

The pressures on our customers often lead them to ask MITRE for quick-reaction responses. To the extent that a quick response is practical, we must provide it. (When the imposed constraints make an informed response impractical, we need to define the extent to which we can make an informed response, explain why we cannot go further, and refuse the remainder of the task.) Our processes for identifying and leveraging applicable past analyses and data, informed professional judgments, and relevant experiences (either within or external to MITRE) need to be focused on enabling the highest quality response within the constraints imposed. Whenever possible, we should document our delivery (even after the fact)—the assumptions made, the methods used, and the results conveyed. We also must develop our knowledge base to continually improve our ability to respond to future requests related to our core competencies.

Moreover, we must assess the risks of quick responses to understand the possible issues with their accuracy and completeness, including the potential consequences of these issues—and so inform the customer. When the risk is high, we should strongly recommend a plan for a more complete, fact-based analysis, using, as needed, trade-space exploration, modeling and simulation, experimentation, proof-of-concept prototyping, etc. Clearly, circumstances requiring in-depth study, especially if associated with key national capability outcomes, demand the highest quality work. This entails careful planning and work shaping, appropriate staffing and resources, peer and management consultation and review throughout the execution of the work, and socializing and delivering the results so that they are correctly interpreted and acted on. It is important to note that the higher level expectations on MITRE can only be met when a significant fraction of our work goes beyond quick response activities, so finding ourselves in these circumstances should be relatively common.

The higher level expectations on MITRE push us beyond responding to customer requests toward proactively identifying key issues on which we can make a difference. These often involve enterprise objectives such as integration and interoperability for information sharing across the government (and, at times, beyond), which may exceed the bounds of an individual customer's purview. When these proactive initiatives lead to substantive efforts, they also demand the highest quality work, applying all the same attributes discussed above to their planning, execution, and delivery.

MITRE needs to provide its customers with "quick and dirty" products when necessary, making them as "clean" as possible but conveying a clean/dirty assessment with the product. Higher level expectations for MITRE's FFRDC contributions often require us to work more substantively, with an even greater emphasis on quality for our work. Quality, then, involves both doing enough of the right work, and doing all of our work (but especially the higher impact work) right. It also includes building relationships so that high impact is, in fact, realized. These objectives are reachable only if we all understand the expectations, are frank and open about assessing the work we're asked to do, foster a culture that values quality and learns from both mistakes and successes, follow through (internally and with customers) on resource allocations, and pay attention to important relationships. Upper management needs to take the lead, but we all need to contribute. Especially with the immediate customer, it's often the project staff that have the frequent connections that influence the customer's perception of our quality and the acceptance of our recommendations.

The Successful MITRE Systems Engineer

What does successful systems engineering look like at MITRE? What is the secret formula for it? As noted early in the companion section to this one—The Evolution of Systems Engineering—there is no single definition of systems engineering and so there is no single definition of success. Much depends on the context in which the systems engineering is being practiced. Nevertheless, the following high-level criteria strongly correlate with successful MITRE systems engineers.

Criteria for Successful MITRE Systems Engineers

Successful MITRE Systems Engineers:

- Define the sponsor's and customer's problem or opportunity from a comprehensive, integrated perspective.
- Apply systems thinking to create strategies, anticipate problems, and provide short- and long-term solutions.
- Adapt to change and uncertainty in the project and program environment, and assist the sponsor, customer, and other stakeholders in adapting to these.
- Propose a comprehensive, integrated solution or approach that:
  - Contributes to achieving the sponsor's, customer's and other stakeholders' strategic mission objectives in a changing environment.

  - Can be feasibly implemented within the sponsor's and customer's political, organizational, operational, economic, and technical context.
  - Addresses interoperability and integration challenges across organizations.
  - Shapes enterprise evolution through innovation.
- Cultivate partnerships with our sponsors and customers to work in the public interest.
- Bring their own and others' expertise to provide sound, objective evidence and advice that influences the decisions of our sponsors, customers, and other stakeholders.

Excerpted from the MITRE Systems Engineering Competency Model [7].

References and Resources

1. Office of Management and Budget, November 13, 2009, U.S. Federal Acquisition Regulation (FAR), 35.017.
2. November 21, 2008, DoD Sponsoring Agreement with The MITRE Corporation to Operate the C3I FFRDC.
3. September 25, 2005, Sponsoring Agreement between the FAA and The MITRE Corporation for the Operation of the CAASD FFRDC.
4. February 7, 2008, Sponsoring Agreement Among the IRS and Department of Veterans Affairs and The MITRE Corporation Operating the FFRDC formally known as the Center for Enterprise Modernization.
5. March 3, 2009, Sponsoring Agreement between DHS and The MITRE Corporation to Operate the Homeland Security System Engineering and Development Institute FFRDC.
6. MITRE, May 2008, Systems Engineering Quality at MITRE.
7. MITRE, September 2007, MITRE Systems Engineering Competency Model, ver. 1.

Enterprise Engineering

MITRE

Introduction

Did you ever wonder if your work needs to be enabled to support an international community? Have you anticipated that the security features of your engineering will have to interoperate with other federal agencies or organizations in the same department? Do performance characteristics of capabilities beyond your control impact the performance of your endeavor?

> "Enterprises" are interwoven sets of mission and business endeavors that need to coexist in a rapidly changing and evolving world. MITRE systems engineers (SEs) are expected to bring an enterprise perspective to their activities at whatever scale of the enterprise they operate: subsystem, system, system of systems, or enterprise. SEs should take a comprehensive viewpoint across technical and non-technical aspects of the problem space, and use systems thinking to ask probing questions and trace the implications of potential answers across the enterprise. SEs work with ambiguous issues and partial information to frame the essence of the problem; create strategies that consider all aspects of the problems and needs of the customer, sponsor, and beyond; and engineer scalable, adaptable, and evolvable enterprise solutions that consider the larger stakeholder community.

Background

In the article "Evolving Systems Engineering," MITRE staff considered the topic of "enterprise" definition and came up with the following working definition:

> By "enterprise" we mean a network of interdependent people, processes, and supporting technology not fully under the control of any single entity. In business literature, an enterprise frequently refers to an organization, such as a firm or government agency; in the computer industry, it refers to any large organization that uses computers. Our definition emphasizes the interdependency of individual systems and even systems of systems. We include firms, government agencies, large information-enabled organizations, and any network of entities coming together to collectively accomplish explicit or implicit goals. This includes the integration of previously separate units. The enterprise displays new behaviors that emerge from the interaction of the parts [1].

MITRE works on projects supporting specific customer needs and their required capabilities. To be successful, MITRE staff must also understand the enterprise context associated with these specific activities. Our customers truly value the enterprise perspective we provide. MITRE has worked on our customers' enterprise and specific needs from our inception. With the SAGE [Semi-Automatic Ground Environment] project, we focused early in our history on the needs of the national enterprise for defense and formulated specific radar solutions to implement the required protection. As MITRE has worked on enterprise challenges over time, we've come to realize:

> Enterprise engineering is based on the premise that an enterprise is a collection of entities that want to succeed and will adapt to do so. The implication of this statement is that enterprise engineering processes are more about shaping the space in which organizations develop systems so that an organization innovating and operating to succeed in its local mission will—automatically and at the same time—innovate and operate in the interest of the enterprise. Enterprise engineering processes are focused more on shaping the environment, incentives, and rules of success in which classical engineering takes place. Enterprise engineering coordinates, harmonizes, and integrates the efforts of organizations and individuals through processes informed or inspired by natural evolution and economic markets. Enterprise engineering manages largely through interventions instead of controls [2].

Major topics and considerations for MITRE staff engineering enterprise solutions are:

- Taking a comprehensive viewpoint
- Enterprise planning and management
- Enterprise technology, information, and infrastructure
- Addressing the complex issues associated with information-intensive environments
- Engineering systems for mission assurance
- Transformation planning and organizational change
- Understanding the enterprise's governance operations along with related assumptions and constraints
- Independent engineering assessments

Comprehensive Viewpoint

A comprehensive viewpoint helps the MITRE engineer create a solution that considers and accounts for the many factors associated with an advantageous path across an enterprise and the environment where the enterprise must operate. There are many complexities to assess and negotiate as we evaluate a comprehensive perspective of the solution space. MITRE engineers can apply a variety of tools to help gain an understanding of the uncertain environment that affects their enterprise. Articles in this topic area include "Systems Thinking," "Systems Engineering Strategies for Uncertainty and Complexity," and "Tools to Enable a Comprehensive Viewpoint."

Enterprise Planning and Management

Enterprise planning and management takes a strategic view of the major plans and processes needed for a federal government organization to achieve its mission. The legislative branch does not often get into details about which components of an executive branch agency will execute each aspect of the mission, or how they will operate. Therefore, at the strategic level, each agency must plan, manage, and account for both how and to what extent it achieves that mission. MITRE engineers are sometimes asked by sponsors to help develop and execute these strategic-level plans and processes. Articles in this topic area include “IT Governance,” “Portfolio Management,” and “How to Develop a Measurement Capability.”

Enterprise Technology, Information, and Infrastructure

The term “enterprise technology, information, and infrastructure” refers to the concept of information technology (IT) resources and data that are shared across an enterprise. Embodied in this concept are technical efforts such as infrastructure engineering for building, managing, and evolving shared IT; IT or infrastructure operations for administering and monitoring the performance of the IT service being provided to the enterprise; IT services management; and information services management. Articles in this topic area include “IT Infrastructure Engineering,” “IT Service Management (ITSM),” “Information and Data Management,” and “Radio Frequency Spectrum Management.”

Engineering Information–Intensive Enterprises

MITRE’s role in operating systems engineering Federally Funded Research and Development Centers (FFRDCs) places us in an environment where our solutions are predominantly used for information-intensive capabilities. Part of our work program may lead us to hardware or platform considerations for enhancing the capabilities of our customers, but typically the emphasis is on the information needs of the missions and decision makers we support. As such, we need to provide solutions that meet the information needs of our customers:

- Solutions that consider the architectures of the enterprise and how to federate the elements to provide integrated capabilities
- Solutions that consider the complexity of the comprehensive viewpoint and formulate approaches to take advantage of design patterns and agile techniques while planning an evolutionary strategy to satisfy the longer term enterprise needs
- Solutions that can be created on-demand for the particular challenge at hand using available resources such as open system capabilities while meeting the rapidly changing and real-time events of the nation

Articles in this topic area include “Architectures Federation,” “Design Patterns,” “Composable Capabilities On Demand (CCOD),” “Open Source Software (OSS),” and “Privacy Systems Engineering.”

Systems Engineering for Mission Assurance

The concept of engineering a system that can withstand purposeful or accidental failure or environmental changes has a long history in the discipline of designing systems for survivability. In the Internet era, engineering systems for mission assurance has been further expanded to include engineering for information assurance and cyber security. In this guide, the definition of “systems engineering for mission assurance” is the art of engineering systems with options and alternatives to accomplish a mission under different circumstances and the capability to assess, understand, and balance the associated risks. Options and alternatives will normally take the form of a blend of technical and operational elements, which requires the systems engineer to have an intimate understanding of the technical details and limitations of the system, the doctrine and operations for its use, and the environmental conditions and threats that will or may be encountered. Taken together, the various dimensions of mission assurance pose some of the most difficult challenges in engineering systems today. The systems engineering community does not yet have complete answers to its myriad questions.

The articles in this topic are focused on what we know about systems engineering for mission assurance today. It is a rapidly evolving field, so check back often for updates and additional material. Articles in this topic area include “Cyber Mission Assurance,” “Crown Jewels Analysis (CJA),” “Cyber Threat Susceptibility Assessment,” “Cyber Risk Remediation Analysis,” “Secure Code Review,” and “Supply Chain Risk Management.”

Transformation Planning and Organizational Change

Transformational planning and organizational change is the coordinated management of change activities that enable users to adopt a new vision, mission, or system. MITRE systems engineers assist in formulating a strategy and plans, and in leading and communicating change. Articles in this topic area include “Performing Organizational Assessments,” “Formulation of Organizational Transformation Strategies,” “Stakeholder Assessment and Management,” “Effective Communication and Influence,” and “Planning for Successful User Adoption.”

Enterprise Governance

MITRE engineers need to understand the mechanisms used by the government to "govern" systems engineering and the capabilities required to accomplish the tasks of the enterprise.

> Governance is the activity of governing. It relates to decisions that define expectations, grant power, or verify performance … governance relates to consistent management, cohesive policies, processes and decision-rights for a given area of responsibility [3].
>
> IT Governance primarily deals with connections between business focus and IT management. The goal of clear governance is to assure the investment in IT general business value and mitigate the risks that are associated with IT projects [4].

Governance engineering requires MITRE staff to work on the social engineering and social networking aspects of systems engineering by using, and sometimes working around, the governance structures. Governance in this area is defined as where the interdependent people, processes, and technology come together to accomplish the required actions to implement the needs of and evolve the enterprise.

Articles in this topic area include "Communities of Interest and/or Community of Practice," "Standards Boards and Bodies," and "Policy Analysis."

MITRE FFRDC Independent Assessments

MITRE systems engineers perform many types of independent assessments, which are known by various names including independent reviews, red teams, appraisals, audits, and compliance assessments. Very often independent assessments are done to identify risks to a program. They provide value to government organizations because the MITRE FFRDC role promotes independence, objectivity, freedom from conflicts of interest, and technical expertise. Related to Contractor Evaluation, this topic area includes the article "Planning and Managing Independent Assessments."

Other Enterprise Engineering Articles

In the future, any articles on subjects of relevance to enterprise engineering but that don't neatly fit under one of the section's existing topics will be added in a separate topic, Other Enterprise Engineering Articles. Such articles are likely to arise because the subject matter is at the edge of our understanding of systems engineering, represents some of the most difficult problems MITRE systems engineers work on, and has not yet formed a sufficient critical mass to constitute a separate topic.

References and Resources

1. The MITRE Corporation, 2011, Evolving Systems Engineering.
2. Rebovich, G., March 2007, Engineering the Enterprise.
3. Wikipedia contributors, “Governance,” Wikipedia, accessed January 27, 2010.
4. Smallwood, D., March 2009, “IT Governance: A Simple Model,” ebiz.

Enterprise Engineering Contents

Comprehensive Viewpoint

Definition: *A broad understanding of the context and environment in which the systems engineering activity or problem is embedded and to be solved. A comprehensive viewpoint enables the ability to develop solutions that consider all aspects of a problem, their relationships and interactions, including current and future needs of the user, customer, and sponsor as well as political, organizational, economic, operational, and technical issues.*

Keywords: *agility, complexity, domain, enterprise, systems, systems thinking, tools, users*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to develop a broad understanding of their problem context and environment. They should consider current and future needs of the sponsor, customer, and operational user, and take into account political, organizational, economic, operational, and technical aspects of the problem and its potential solutions. They are expected to use this comprehensive view to develop, recommend, and lead systems engineering activities in the enterprise. In doing so, MITRE SEs consider:

- Operational needs and the changing global environment that the nation and our operational users must work within, including the collection of systems with which our individual projects interact

- Technical environment, its rapid evolution and how it influences feasible implementation approaches
- Economic constraints and processes that influence solutions and their implementation
- Agendas and perspectives of the stakeholder community (in the customer chain and across the mission and domain areas)
- International partners and the policies that govern how we work in the international community
- Data and information needs, processing, security, and applications that are required to get results.

Comprehensive Viewpoint: The Sponsors' Requirement

As a corporation that operates federally funded research and development centers (FFRDCs), MITRE is required to take a comprehensive viewpoint of all of our work. This requirement is specifically delineated in the individual FFRDC sponsoring agreements, as shown in the following excerpts:

- **Department of Defense (DoD) Command, Control, Communications and Intelligence (C3I) FFRDC Sponsoring Agreement:** "While serving the immediate needs of the many individual programs it supports, the C3I FFRDC aligns its work program to assist in achieving integrated enterprise capabilities [1, p. 3]."
- **Federal Aviation Administration (FAA) Sponsoring Agreement:** "CAASD [Center for Advanced Aviation System Development] is … to solve problems that are too broad and too complex to … stand alone but are so linked to others that a highly specific analysis may be misleading [2, p. 5]."
- **Center for Enterprise Modernization (CEM) Sponsoring Agreement:** "… simultaneously direct its efforts to the support of individual programs and projects for enterprise modernization, assuring that these individual programs and projects operate effectively with one another and efficiently support the … objectives of the Government [3, p. 3]."
- **Homeland Security Systems Engineering and Development Institute [SEDI] Sponsoring Agreement:** "… shall promote compatibilities across the various homeland security platforms and equipment … through, among other things, improved interoperability and information sharing within the homeland security enterprise [4, p. 2]."

Comprehensive Viewpoint: Leveraging the Corporation

MITRE's sponsoring agreements not only direct us to take a comprehensive viewpoint across the sponsor's enterprise, but extend across all of our FFRDCs to ensure we are formulating national solutions to hard problems. The following excerpts from the CEM sponsoring agreement illustrate this. The other sponsoring agreements contain similar language.

> "... ensure that the FFRDC's work programs can be accomplished in a complementary manner that draws on the entire range of corporate competencies [3, pp. 3-4]."
>
> "[MITRE's several FFRDCs] ... are operated in such a way as to enhance the technical quality and objectivity of each [3, p. 3]."

Within MITRE, we often refer to this requirement as "bringing the corporation to bear." It is for this reason that MITRE emphasizes collaboration and networking across the corporation. More recently, we have extended this concept to "bringing the world to bear," by which we emphasize collaboration beyond our corporate boundaries to wherever the greatest expertise to solve a problem resides—other FFRDCs, academia, industry, and international partners.

Articles Under This Topic

The article "Systems Thinking" provides a general introduction to the art and practice of examining the totality of a problem, including the environment in which the problem is contained, as well as the linkages and interactions among the problem's parts. Systems thinking is used in problems in which cause and effect are not closely related in space or time, as well as problems in which the relationships among elements are nonlinear. Systems thinking enables alignment of purposes, which is so important to successful engineering of enterprise capabilities because it enables the systems engineer to ask purposeful questions and trace the implications of potential answers across their enterprise.

Increasingly, the complexity we encounter in the enterprises and systems that MITRE helps engineer requires a spectrum of systems engineering techniques. When a system is bounded with relatively static, well-understood requirements, the classical methods of systems engineering are applicable and powerful. At the other end of the spectrum, when systems are networked and each is individually reacting to technology and mission changes, the environment for any given system becomes essentially unpredictable. The article "Systems Engineering Strategies for Uncertainty and Complexity" discusses the nature and sources of uncertainty in engineering IT-intensive, networked systems and suggests strategies for managing and mitigating their effects.

There are a variety of cognitive tools to help apply a systems thinking perspective to the increasingly complex problems MITRE encounters. The article "Tools to Enable a Comprehensive Viewpoint" describes a set of tools to help MITRE systems engineers understand and characterize the nature and source of uncertainty and complexity in their environment.

Best Practices and Lessons Learned

Look for opportunities to contribute to solving the broader integration and interoperability challenges across your enterprise at the same time you solve your particular project's problems.

As you do your day-to-day work, keep your head up to understand where and how your particular activity fits into the larger context.

Understand MITRE's systems engineering quality construct [5], and use it to guide the execution of your work activities.

Recognize and act on the understanding that a locally optimal solution for a problem may be suboptimal for the enterprise and less advantageous overall than other solutions. For example, working a data strategy across a broader community may preclude a more elegant solution to a particular system application, but the increased value of data sharing and interoperability across the broader community outweighs the benefits of a program-centric solution.

References and Resources

1. November 21, 2008, DoD Sponsoring Agreement with The MITRE Corporation to Operate the C3I FFRDC.
2. December 22, 2010, Sponsoring Agreement Between the FAA and The MITRE Corporation for the Operation of the CAASD FFRDC.
3. February 7, 2008, Sponsoring Agreement Among the IRS and Department of Veterans Affairs and The MITRE Corporation Operating the FFRDC formally known as the Center for Enterprise Modernization.
4. March 3, 2009, Sponsoring Agreement between DHS and The MITRE Corporation to Operate the Homeland Security System Engineering and Development Institute FFRDC.
5. Metzger, L., May 2009, Systems Quality at MITRE, The MITRE Corporation.

Additional References and Resources

"Comprehensive Viewpoints," MITRE Systems Engineering Competency Model, The MITRE Corporation.

"FFRDC Role and Public Interest," MITRE Project Leadership Handbook, The MITRE Corporation.

COMPREHENSIVE VIEWPOINT

Systems Thinking

Definition: *The ability and practice of examining the whole rather than focusing on isolated problems (P. Senge) [1]. The act of taking into account the interactions and relationships of a system with its containing environment (Y. Bar Yam, New England Complex Systems Institute).*

Keywords: *holism, holistic, interactions, multidimensionality, multiple perspectives, relationships, synthesis, synthetic, system thinking, systems thinking*

MITRE SE Roles and Expectations: MITRE systems engineers are expected to: (a) understand the linkages and interactions among the elements of their system or enterprise and its connecting entities; (b) align goals and purposes across the enterprise; and (c) ask probing questions and trace the implications of potential answers across the enterprise.

Background

The recently renewed interest in systems thinking in government circles and the engineering community has been fueled, in part, by the movement to apply systems science and complexity theory to problems of large-scale, heterogeneous, information technology-based systems.

Systems thinking is a framework for solving problems based on the premise that a component part of an entity can best be understood in the context of its relationships with other components of the entity, rather than in isolation. The way to fully understand why a problem occurs and persists is to understand the "part" in relation to the "whole." A focus of systems thinking is on understanding the linkages and interactions among the elements that compose the entirety. Systems thinking is often used in problems in which cause and effect are not closely related in space or time, as well as problems in which the relationships among components are nonlinear (also see the SEG article "Systems Engineering Strategies for Uncertainty and Complexity").

Systems thinking requires knowledge and understanding—both analysis and synthesis—represented in the same view. The ability to combine analytic and synthetic perspectives in a single view enables alignment of purposes, which is so important to successful engineering of enterprise capabilities. It allows the systems engineer to ask purposeful questions and trace the implications of potential answers across the enterprise. Would a change in performance at the subsystem level result in a change at the enterprise level? If so, how, and is it important? How would a new enterprise-level need be met?

The following concepts are important in applying systems thinking:

- **Analysis:** The ability to decompose an entity into deterministic components, explain each component separately, and aggregate the component behaviors to explain the whole. If the entity is a system, then analysis answers the question, "How does the system work?" Analysis results in knowledge of an entity; it reveals internal structure. For example, to know how an automobile works, you analyze it—that is, you take it apart and determine what each part does. This is essential to important activities like repairing automobiles or diagnosing and repairing problems of other, more complicated systems.
- **Synthesis:** The ability to identify the whole of which a system is a part, explain the behavior or properties of the whole, and disaggregate the whole to identify the role or function of the system in the whole. Synthesis answers the "Why is it what it is?" question. Synthesis is the mode of thought that results in the understanding of an entity (i.e., an appreciation of the role or function an entity plays in the larger system of which it is a part). For example, the reason why the American automobile was originally designed for six passengers is because the average family size at the time was 5.6. Every MITRE systems engineer who has defined a system performance specification against mission or operational requirements has used synthetic thinking.

Example

To analyze an unmanned aerial vehicle (UAV), we would logically decompose it into subsystems that perform certain roles or functions (e.g., the host platform, sensors that perform ground target detection and identification, communication systems for receiving commands from controlling entities, transmitting onboard sensor data to control systems, etc.) down to a level of granularity sufficient to answer the question at hand.

To understand the UAV system synthetically, the first step is to identify the larger system of which the UAV is a part, (e.g., a situational awareness [SA] system of systems [SoS]). The second step is to describe the containing system (e.g., the SA SoS delivers high-quality location and identification information on militarily significant objects of interest in a surveillance volume, with an emphasis on ground, sea surface, and low-altitude air vehicles). The third step is to disaggregate the whole to identify the role or function in the larger system of which the UAV is a part (e.g., the organization employing the SA SoS has a mission that focuses on the detection, location, and identification of ground and sea surface vehicles. The UAV in question is equipped with sensors and processing tailored to ground vehicle detection.). Taken together, this synthetic view explains why the organization has ground vehicle detection UAVs in its SA SoS and provides a basis for asking and answering "what if?" questions about the UAV, like: "What if the organization's mission shifted away from ground vehicle detection or moved more toward it?"

Combining the analytical and synthetic perspectives in a single view allows the systems engineer to ask questions and draw implications of potential answers across the enterprise. If the organization's mission shifted to ultra-light airborne vehicle detection, how would SA be accomplished? Could the existing UAVs be re-engineered or refitted with new sensors to detect and identify the new target types? Would a change in performance at the UAV system level result in a change at the SA SoS or mission level? If so, how, and is it important?

Government Interest and Use

The need to apply systems thinking continues to be pervasive across MITRE. It is expected of MITRE by our sponsors. Reference to it is made in our sponsoring agreements:

- **From the FAA Sponsoring Agreement:** "CAASD is uniquely qualified...to solve problems that are too broad and too complex to become the focus of a competitive procurement..."
- **From the DoD Sponsoring Agreement:** "While serving the immediate needs of the many individual programs it supports, the C3I FFRDC aligns its work program to assist in achieving integrated DoD-wide enterprise capabilities..."
- **From the IRS Sponsoring Agreement:** "The FFRDC shall simultaneously direct its efforts to the support of individual programs and projects for tax modernization, and to

> assuring that these individual programs and projects operate effectively with each other and efficiently support the business objectives of the government…"

Systems Thinking Best Practices

Systems engineering and systems thinking have always been about asking good questions and forming conclusions and recommendations based on the answers. When performing your MITRE systems engineering activities, consider asking these sorts of questions:

What is my enterprise? What elements of it do I control? What elements do I influence? What are the elements of my environment that I do not control or influence but which influence me? [2, pp. 2-3 – 2-4]

Can a balance be achieved between optimizing at the system level and enabling the broader enterprise? If the balance comes at the expense of the smaller system, can that be offset or mitigated? How?

Is interdependence of performance measures (variables) in a system or enterprise hidden by slack? Is the inability to make progress in one measure, except at the expense of others, an indication that the slack among them has been used up? Can a redesign of the system or enterprise remove interdependence or provide additional slack? [2, pp. 4-9 – 4-10]

How can analytic and synthetic perspectives be combined in a single view to enable alignment of purposes across the enterprise? Would a change in performance at the subsystem level result in a change at the enterprise level? If so, how, and is it important? How would a new enterprise-level requirement be met and how would it influence its constituent systems?

Can the solution space of a seemingly intractable problem be expanded by viewing it in its containing whole? How? [2, pp. 4-3 – 4-4]

References and Resources

1. Senge, P., et al., 1994, *The Fifth Discipline Fieldbook*, New York, NY, Doubleday.
2. Rebovich, G., 2005. Systems Thinking for the Enterprise: New and Emerging Perspectives, The MITRE Corporation.

Additional References and Resources

Ackoff, R., November 1993, "From Mechanistic to Social Systemic Thinking," System Thinking in Action Conference.

Axelrod, R., and M. D. Cohen, 2000, *Harnessing Complexity: Organizational Implications of a Scientific Frontier*, New York, NY, Basic Books.

Gharajedaghi, J., 1999, *Systems Thinking: Managing Chaos and Complexity*. Boston, MA, Butterworth Heinemann.

Rebovich, G., 2006, “Systems Thinking for the Enterprise: A Thought Piece,” International Conference on Complex Systems, Bedford, MA, The MITRE Corporation.

Rebovich, G., 2006, “Systems Thinking for the Enterprise: New and Emerging Perspectives,” *Proceedings of 2006 IEEE International Conference on Systems of Systems.*

COMPREHENSIVE VIEWPOINT

Systems Engineering Strategies for Uncertainty and Complexity

Definitions: *External uncertainty includes changes in the market, the operating environments, business processes, and threats. Internal uncertainties include program/project execution as well as design, implementation, and performance challenges [1]. Complexity is the interactions and interdependencies among people, organizations, technologies, tools, techniques, procedures, and economics that create patterns that transcend the goals of any one group. Complex interactions can result in resilience and robustness but also in cascading failures [2, 3].*

Keywords: *adaptability, agility, complex systems, complexity, ecosystem, emergent behavior, fitness, flows, interactions, interdependency, robustness, selection, uncertainty, variety*

MITRE SE Roles and Expectations: MITRE systems engineers are expected to understand the nature and sources of uncertainty, lack of effective control [4], and complexity [5] in their environment and then select and apply appropriate strategies for managing or mitigating their effects.

Introduction

The complexity we are seeing in the enterprises and systems that MITRE helps engineer requires a spectrum of systems engineering techniques. When a system is bounded with relatively static, well-understood requirements, the classical methods of systems engineering are applicable and powerful. At the other end of the spectrum, when systems are networked and each is individually reacting to technology and mission changes, the environment for any given system becomes essentially unpredictable.

The metaphor of the watchmaker and gardener is sometimes used to describe the differences between engineering in the two types of environments [6]. Classical systems engineering is like watchmaking. Its processes, techniques, and tools are applicable to difficult problems that are essentially deterministic or reductionist in nature. Like gardening, engineering for the enterprise draws on the fundamental principles of evolution, ecology, and adaptation. It uses techniques to increase the likelihood of desirable or favorable outcomes in complex environments that are characterized by uncertainty and that may change in unpredictable ways. Engineering for the enterprise is not a replacement for classical systems engineering. Increasingly, both disciplines must be used in combination to achieve success.

This article begins with a discussion of ecosystems and includes a large number of footnotes and references for the interested reader. This will strike some as an odd place to start. But in many ways the point of view required to understand ecology is analogous to the one needed to comprehend the complex environment in which MITRE systems engineers find themselves today. In fact, a number of the emerging best practices and lessons learned discussed later in this article draw on ecology or evolutionary biology for their inspiration. Last, the best practices and lessons learned are organized around important conundrums, needs, or issues that systems engineers face in complex environments.

Because engineering in complex environments is an emerging and rapidly changing field, MITRE systems engineers and others are developing its processes, techniques, and tools as they execute their program responsibilities. As a result, in many cases, there is an inherent uncertainty about the right wisdom to recommend. But pointing them out has value even if we don't yet know exactly the right wisdom to convey about solving them. When it has been possible to suggest at least potential approaches to dealing with a problem, the article does so.

Background

People exist within an ecosystem. We have a sense of what that means and understand the natural world around us as elements in an ecosystem. Our technical systems exist within ecosystems as well. We need to unwrap what it means to be systems engineers within an ecosystem; and, thus, understand the nature and sources of uncertainty, lack of control, and complexity in our environment [7].

Most people have a keen sense of what a natural ecosystem consists of, and how it morphs over time in response to changes of (and to) its constituent members. Ecosystems are dynamic. Their aggregate state is arrived at through various interactions among the elements present (some connected directly, others indirectly through some transitive path), and how the elements push and pull all the time along many dimensions. Any apparent stability is a dynamic stability which, when one of the interacting elements is altered, changes the stability point of the aggregate—and the changes ripple through the connected pieces (sometimes rather rapidly—and unexpectedly) until another dynamic stability point is found.

Ecosystems are distributed and also have no leader; no one's "in charge." This is nothing that needs to be "fixed"—in fact, it can't be fixed [8].

All of the systems we work on at MITRE have always existed in this type of ecosystem and have been subject to this type of push-and-pull. But three things have changed:

1. In the past we have used people as the "impedance matching" element between the artificial elements (traditionally, fully formed, and conceived systems); now artificial elements are connecting to other artificial elements (machine to machine).
2. The wide potential interconnections we now accept (and demand) among the artificial elements (composition on demand [9]).
3. The engineering we are now expected to perform at large scopes and scales (enterprise engineering).

We now find our systems to be primary elements of the ecosystems in which they reside, rather than augmentations to the primary elements (i.e., the people using them), and we must factor that into our requirements, analyses, and designs [10]. They must be able to respond to changes in the context they find themselves within, rather than relying on people to be the elements that change in response to context changes (i.e., the environment).

Note also that this environment is changing at rapid and unpredictable rates, and in places we didn't necessarily predict. The technology itself is also changing at unprecedented rates. Thus we are finding that agility is most desired. The systems themselves must be agile; not just the users of the systems [11, 12]. Most important, isolation (or attempted isolation) doesn't work.

Having made the argument for variety and interaction, it is important to add the guiding factor: selection. Arbitrary and random change merely leads to chaos. However, the environment guides or channels change by selecting the winners and the losers among those present. Those chosen are said to be "more fit" for the environment. Thus fitness, and its measurement, might be something to pursue [13].

Given multiple interdependencies, rippling change, an unknown (and possibly unknowable) future, and selection among choices, then, clearly, we can expect uncertainty and therefore agility is a top need. But agility of what?

- **Agility of the aggregate:** "Systems" and "systems of systems" are nothing more than collections of technical elements, people, and processes that perform useful work. It is in the aggregate where this useful work is realized. To be agile, the aggregates must be able to adapt (and possibly, to be assembled) near the point of need, and in a timeframe that allows the potential solution (the aggregate) to be applied to the need in a timely way.
- **Agility of the elements:** Each element within an aggregate must itself be able to evolve. The rate of its evolution—or its ability to change rapidly with the emergence of a new need—may well define its value to the enterprise. Thus adaptability becomes a strong design aspect.

It is within this environment, and with these challenges, that MITRE systems engineers are expected to perform. One needs to have this understanding and mindset. It is within this mindset that we can see users as arbiters of "fitness" and the greater stakeholder community as the environment [14].

Government Interest and Use

The government has a direct interest in seeing that systems built are agile and composable in order to meet the changing ecosystem in which our government customers live. Examples of capabilities MITRE has built this way are in the SEG article, "Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation." Being agile and composable satisfies the ability to change quickly as conditions, technologies, missions, and procedures change. It also suggests that we may be able to achieve more (re)usability and thus more effectively manage cost. Uncertainty becomes less of a problem if agility is possible. It allows rapid reaction to current conditions rather than prediction of future conditions followed by subsequent reaction/change. Best practices and lessons learned fall along these lines [15, 16, 17, 18, 19].

Best Practices and Lessons Learned

Technology

Given an unknown future, MITRE systems engineers are expected to consider and recommend the value of building options into designs [20]. They are expected to envision possible system or enterprise extensions in advance, the likelihood of whether and when they would be needed, and the cost of extending the design versus creating a replacement.

Partition design by both functionality and time differences of change. Traditional design tends to partition primarily by function. However, partitioning also by rate of change allows us to isolate elements that change quickly (or might change quickly) from those elements that are more stable and will change slowly [21].

Encapsulate change. A basic tenet of design that has weathered the test of time is to isolate things that change. Key to this is the use of interfaces as a method to isolate the partitions from each other, yet allow interaction.

Carefully choose "bow ties" [22]. In the design, identify and codify those key decoupling points that divide designs into coherent layers. These should be small in number and ruthlessly enforced. It is the essence of workable architectures. A small number of connection/decoupling points of very low variety (i.e., goal of one) allows high variety on each side of these strategic points. The key decoupling points should use well-known and popular protocols and methods to ensure they have "legs."

Building an enterprise element while building a local system. Understand your offering to the enterprise:

- What does it do (the single thing that provides value to the enterprise)?
- How do others interact with it?
- Where/how is it available?
- How do others find it?

Refactoring for the enterprise. Once local elements are discovered and used by the enterprise (i.e., by consumers outside of those originally anticipated by the program originators), refactoring their appearance and presentation to the enterprise is likely warranted. This could mean:

- Splitting a system into two or more (allowing each part to change at its own rate, or permitting access and interaction to only a piece of the original whole).
- Substituting one element for another (allowing a new element to perform a role previously provided by another). This allows evolution and change and is the fundamental idea behind interface implementer substitution.
- Augmenting a system with new elements (adding on new elements may allow new roles for the system).
- Inverting element dependencies to alter business/political considerations (consider the different political/business dynamics resulting from using a design pattern such as subclass/inheritance vice containment/delegation).

The actions in the previous bullets have been argued to be design primitives [23].

Flows [24] and their emergence. Information flows are the essence of command and control systems. Often we used defined flows in the past within our designs to decide what elements in a system need to connect together to realize a system's behavior. To achieve agility, however, we need to create designs that allow technical elements to join and leave existing flows dynamically, and which will enable the creation of new flows.

Structure and Organization

MITRE systems engineers are expected to consider, recommend, and apply systems engineering strategies such as early prototyping, exploratory integration test-beds, field trials, and experiments to support early and continuous discovery activities in situations in which the required behavior of the deployed system(s) is difficult to predict.

- **Development networks.** Mimic the real world as much as possible.
    - Providing vetted access to online-available software services that are also found in the fielded system allows third parties to learn about and use aspects of the system-of-record that would otherwise need to be guessed at.
    - Third-party developers who use the resources available on the development network will require less integration, hand-holding, and rework, thus speeding fielding and holding costs under control.
- **Developmental spirals.** Because the future is difficult to predict, using spirals (smaller scope, shorter duration) to sharpen the focus on future requirements lowers uncertainty and risk.
- **Modeling and simulation.** People are poor at predicting patterns formed from the interactions of elements (e.g., rules, computing artifacts, etc.). The only way that we may fairly, and without introducing additional bias, elicit patterns (other than the choices and assumptions that go into a model, which should be explicit) is to use modeling and simulation to explore the interactions (be they operational, technical, or systemic).

Piloting integration strategies. MITRE systems engineers are expected to consider, recommend, and implement integration strategy pilots to explore terminology, operational patterns, technology, and desired features when interoperating systems cross multiple seams and lack a history of effectively working together.

Using "technical intimacy"—from casual relations to deep commitment, we are most likely to use (and depend on) an external element when it:

- Already exists.
- Is available.
- Is likely to remain available.
- Is understandable.
- Makes small demands on our resources.
- Requires few unique agreements.
- Appears to be part of the environment.

Replaceability vs. reusability. Focus on designs that offer the ability to replace elements with other (similar) elements as experience is gained with a system, and/or as requirements change, rather than seeking or designing elements that purport to include all future needs. We can start with small sets of known functionality, then grow it. This lowers risk greatly.

Partnerships build trust [25]. Forming partnerships among both consumers and producers of services builds trust. Activity taking place on a development network can provide pointers to potential partnering opportunities that may not have been obvious.

Business and Economic [26, 27, 28]

Reduce uncertainty [29]. MITRE systems engineers are expected to understand the elements that may drive uncertainty in the tasks they're supporting. Uncertainty may come from requirements and/or technologies and MITRE engineers must help customers understand this environment and help mitigate the uncertainty.

- Where a project is stable in both requirements and technologies, we are able to plan ahead, then execute the plan.
- Where the project is dominated by new or emerging technologies, we should consider a strategy of a "portfolio of small bets."
- Where a project is dominated by evolving requirements, we should consider a strategy of "staged commitments."
- Where all characteristics exist, we need a hybrid strategy.

Reduce uncertainty in cost estimation. MITRE systems engineers are expected to understand the principles underlying good cost estimation and be able to recommend and implement techniques to mitigate cost uncertainty, to include developing design alternatives as bases for cost. MITRE's CASA organization has many methods to help MITRE engineers with cost estimating and associated decision analysis.

There are two "truths" in conflict. We need to know what to build before building it, and things always change. Thus the idea that requirements must be known before building is desired, but the requirements themselves may be changing; so, if things always change, knowing what to build may be fuzzy. But "what to build" needs to be known to estimate well.

If it's fuzzy, tighten it up, either in time or scope. Can we define what will be done this year? This month? This week? Find a time slice where this is clear, outcomes are definite, and the method to achieve them is known. Where things become fuzzy, this may well be a point where there's a logical branching of possibilities, and a perfect opportunity for "Real Options" [30] to be developed. This is good for interfaces in which details can be deferred.

With respect to estimation:

- The smaller it is, the easier it is to estimate.
- The simpler it is, the easier it is to estimate.
- The more mature the technology is, the easier it is to estimate.
- The more that is supplied by others, the less needs to be done (i.e., the smaller it is).

There are many approaches we can take for an estimation methodology. They all share one key characteristic: none is able to satisfy all. This goes from agile and lean techniques [31], which measure team velocity delivering story points, to function points, and the classic SLOC (source line of code) counts. Be very wary of whatever technique is chosen. Don't automatically accept it—always seek supporting and refuting evidence on the estimates during execution.

Establishing baselines. The baselines should be appropriate for the estimation method and the development measurement methods. For example, "done done" in agile methods should be ruthlessly watched. This fits well with defining earned value milestones (EVM) [32]. A potential benefit of EVM is that it demands a crisp definition of a milestone and provides early hints when the cost and schedule assumptions are being violated. This may provide a tool for knowing when to abandon one option and pick another.

A hidden problem with service-oriented approaches [33]. Ironically, although service-oriented approaches offer the potential agility and composability desired, the manner in which we contract with developers may erect barriers to realizing the benefits. Consider the situation in which a program offers a service that delivers some of its information bundled in a collection. Suppose further that this is discovered and found useful by many outside the originally planned users and stakeholders. Under these circumstances, we might expect the use of the service to be greatly beyond the planned use. However, transaction densities may exceed the design limits and degrade the performance. Whose problem is this, and how is it mitigated?

Contract types. Consider using contract structures for which the default behavior is not continuation. We might do this using an indefinite delivery/indefinite quantity contract with a series of tasks that build on one another, yet where each has a clear ending.

Consider using "supplier" models in contrast with "developer" models. Payout based on use. There are many challenges to working the uncertainty and complexity of MITRE's customer environment. The need to manage these challenges has become more prevalent in the tasks MITRE takes on as we continue to work the strategic and hard problems of our customers and their enterprises. The practices listed can help work this critical area—as more experience is gained by MITRE staff, these practices will evolve as well in our uncertain and complex MITRE world.

References and Resources

1. As characterized in Stevens, R., September 24, 2009, "Acquisition Strategies to Address Uncertainty: Acquisition Research Findings," from MITRE-Sponsored Research, "Enterprise Systems Acquisition Using Venture Capital Concepts."
2. Dorner, D., 1996, *The Logic of Failure*, Basic Books.
3. Mitchell, M., 2009, *Complexity: A Guided Tour*, Oxford.
4. "Lack of control" includes many conditions and situations, but the most general sense of its use here is the inability to set the desired state, or vector, of an element under one's authority, and which one is expected to exercise control over. The "traditional" approach to ensuring control is isolation of a system—both in development and in use. With more interconnected and interdependent elements and systems, this presumption (and technique) is violated.
5. "Complex" has become a term of art in engineering and science, and its meaning is slightly different than how it is used in the vernacular. At the risk of oversimplifying, "complicated" as a word means difficult to understand, whereas "complex" means stable

collections and patterns arising from (or emerging) from simple interactions among component pieces. See almost any of the references for more description.

6. Metzger, L. M., April 27, 2009, "MITRE Systems Engineering: So What's It All About?"
7. Bar-Yam, Y., 2004, *Making Things Work: Solving Complex Problems in a Complex World*, Knowledge Press.
8. Johnson, S., 2001, *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*, Scribner.
9. Also see the SE Guide article "Composable Capabilities On Demand (CCOD)" in Engineering Information-Intensive Enterprises.
10. DeRosa, J. K., et al., 2008,"The Requirements Conundrum in Complex Systems Engineering," ICSSEA 2008, Paris.
11. Watts, D., 2003, *Six Degrees: The Science of a Connected Age*, Norton.
12. Barabasi, A-L., 2002, *Linked: The New Science of Networks*, Perseus.
13. Perhaps only the recognition of fitness as a process is sufficient, and the understanding and management of choices and "choice-spaces" is where we can make our engineering processes reflect fitness. If we were to be able to understand and quantify fitness, it might give us a predictive tool that is currently absent.
14. There are many efforts that attempt to quantify fitness in engineering systems. In our own community, there are attempts to define measures of effectiveness focusing on operational metrics.
15. Norman, D. O., and B. E. White, 2008, "Asks the Chief Engineer: 'So What Do I Go Do?!,'" SysCon 2008, IEEE International Systems Conference.
16. Norman, D. O., and M. Kuras, 2006, "Engineering Complex Systems," *Complex Engineered Systems*, Springer, Chapter 10.
17. Norman, D. O., 2009, "Engineering Complex Systems: challenges in the theory and practice," *Organizing for a Complex World*, CSIS Press, Chapter 7.
18. Friedman, T. L., 2006, *The World Is Flat,* Farrar, Straus, and Giroux.
19. Ackoff, R. L., F. Emery, 2006, *On Purposeful Systems*, Transaction.
20. Options provide variety. And, variety is absolutely necessary to promote evolution. This may seem counterintuitive to those who view variety as mere redundancies. It must be recognized that variety (also diversity) requires selection to lead toward effective evolution. Variety is explained nicely by Ross Ashby in "Law of Requisite Variety" in his book *Introduction to Cybernetics*, 1956, Chapman and Hall, London. Also see *Diversity Collapse: Causes, Connections, and Consequences*.

21. Think about automobiles. If we didn’t allow for the removal and replacement of the wheels/tires, we would need to wait for a different (redesigned) vehicle in order to operate the vehicle in a different environment—like loose sand rather than asphalt. By recognizing that wheels/tires can be changed quicker than the vehicle is replaced, we allow change at that point, and the evolution of the whole can occur more rapidly.
22. Also see the SE Guide articles, “Architectures Federation” and “Design Patterns” in Engineering Information-Intensive Enterprises.
23. Baldwin, C., and K. Clark, 2000, *Design Rules: The Power of Modularity*, Vol. 1, MIT Press.
24. Holland, J., 1995, *Hidden Order: How Adaptation Builds Complexity*, Perseus.
25. Moore, J. F., 1996, *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*, Harper Business.
26. Beinhocker, E., 2006, *The Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics*, HBS Press.
27. Wheatley, M. J., 1999, *Leadership and the New Science: Discovering Order in a Chaotic World*, Berrett Koehler.
28. Also see the SEG topic Acquisition Program Planning in Acquisition Systems Engineering Section.
29. Stevens, R., September 24, 2009, “Acquisition Strategies to Address Uncertainty: Acquisition Research Findings,” from her MITRE-Sponsored Research “Enterprise Systems Acquisition Using Venture Capital Concepts.”
30. Real Options research at MITRE.
31. Shore, J., and S. Warden, 2008, *The Art of Agile Development*, O’Reilly.
32. Fleming, Q., and J. Koppelman, 2005, *Earned Value Project Management*, 3rd ed., Project Management Institute.
33. Martin, J., 1995, *The Great Transition*, Macon.

COMPREHENSIVE VIEWPOINT

Tools to Enable a Comprehensive Viewpoint

Definition: *A comprehensive view takes a look at a situation and helps describe the complexity of an enterprise and identify the activities necessary to balance interests across potentially competing perspectives throughout the enterprise, such as interconnected mission needs, business requirements, technological enablers, cultural environments, economic constraints, and others. Various tools can be used to formulate a comprehensive view of an enterprise that captures and compares the important drivers, influences, and risks affecting the establishment of desired capabilities.*

Keywords: *comprehensive viewpoint, enterprise, federation, POET, principles, SE Profiler, stakeholder analysis, TEAPOT, tools, value impact, value metrics*

MITRE SE Roles and Expectations: MITRE systems engineers are expected to analyze and understand a customer's enterprise or cross-agency environment in the context of customer and stakeholder needs and challenges. MITRE systems engineers are also expected to formulate and adjust plans and steps needed to effectively provide thought leadership, enhance enterprise integration, identify political challenges, recognize mission/operational gaps, mitigate risks, and ensure delivery.

Enabling a Comprehensive Viewpoint

A comprehensive viewpoint of the customer's environments should portray strengths, weaknesses, challenges, and constraints in all areas pertinent to the work program/project. It is crucial to take a holistic approach when establishing a view of the customer's environments in the context of the intended program/project [1] (also see the SEG article "Systems Thinking"). A well-analyzed and balanced perspective not only provides the facts and information for MITRE systems engineers to devise plans and activities necessary to meet the intended requirements and objectives, it also renders indications for adjustments, improvements, and enhancements [2, 3]. It is important to establish a set of "program basics" to best depict the current state of the working environments as well as associated elements that would assist/ impact the success of the program/project. As starting points for analysis, consider this set of program basics:

- Scope of work program/project
- Work program/project relevant to customer's mission and strategic objectives
- MITRE roles and responsibilities
- Work program/project environments
- Relationships with the customer
- Work program/project management (initiation, planning, execution, and closing)
- Work program status

To adequately portray the current and desired state of the environments, the analyses should be conducted with integrity, objectivity, and consistency. Tools are available for conducting such analyses that can appropriately articulate the states of the customer's environments throughout the program/project life cycle (see Table 1).

Depending on the size and complexity of the program/project, tools can be applied either independently or collectively to describe the strengths, weaknesses, gaps, risks, and issues of the environments being analyzed. Additionally, it is crucial to recognize/identify the interdependencies of the findings to best assist the formulation of the corrective plans and actions. For instance, the root cause of some technical challenges encountered may be the results of deficient stakeholder analyses and ill-defined requirements.

The following tools have been proven useful and effective in analyzing the working environments, devising feasible enhancement/corrective actions, and formulating execution plans and steps.

POET

The Political, Operational, Economic, and Technical (POET) analysis technique was developed by TRW, Inc., in 1999. It was created to assess challenges and opportunities associated with large-scale programs consisting of systems-of-systems. However, it can be used to assess or devise programs, customer challenges, or strategies, regardless of the size and complexity of the program. The analysis uses the POET categories to construct the program basics, identify the program challenges and constraints, and devise action plans accordingly.

- **Political:** Assess and articulate associated leadership, mission/business decision drivers, organizational strengths/weaknesses, policies, governance, expectation management (e.g., stakeholder relationship), program management approach, etc.
- **Operational:** Obtain and evaluate mission capabilities, requirements management, operational utility, operational constraints, supporting infrastructure and processes, interoperability, supportability, etc.
- **Economic:** Review capital planning and investment management capabilities, and assess the maturity level of the associated processes of budgeting, cost analysis, program structure, acquisition, etc.
- **Technical:** Assess and determine the adequacy of planned scope/scale, technical maturity/obsolescence, policy/standards implementation, technical approach, etc.

TEAPOT

The Center for Enterprise Modernization furthers the POET analysis disciplines to promote technical accuracy, economic feasibility, actionable recommendations, political insightfulness, operational reality, and timely delivery (TEAPOT) [4].

In addition to assessing and presenting the challenges and deficiencies, TEAPOT emphasizes the need to define actions and activities to be performed to enhance/improve the current state and to demonstrate the breadth and depth of MITRE's federally funded research and development center role and responsibilities. Here are some examples of TEAPOT application:

- **Technical accuracy:** Use mature technologies and methodologies to assess the soundness of technical requirements and/or solutions; review compatibility among new and legacy systems; determine extensibility and scalability for future changes in scope and requirements, etc.
- **Economic feasibility:** Determine if the total cost of the program/project is within the customer's available funding and proportional to expected benefits; ensure the acquisition/sourcing strategies are adequate, etc.
- **Actionable recommendations:** Present direct and clear recommendations that target identified deficiencies, documented findings, and recommendations objectively and professionally; provide level of detail appropriate for the customer (e.g., executive vs. technical briefings), etc.
- **Political insightfulness:** Recognize the strength and weakness of the organizational culture and socialize findings to ensure understanding and acceptance; make

recommendations that are in compliance with the mandates; balance the competing priorities of key stakeholders, etc.

- **Operational reality:** Consider customer's resource constraints such as staff, systems, funding, etc.
- **Timely delivery:** Plan and deliver on time as scheduled.

Systems Engineering (SE) Profiler

The MITRE-developed Systems Engineering Profiler is used to characterize systems in context and for visualizing system integration problems along multiple dimensions. This tool is particularly useful and effective for programs/projects that involve designing systems that can perform as components of large-scale, complex enterprises. MITRE systems engineers are advised to look beyond the system, and consider the characteristics of the enterprise in which the system will function and the context in which the system is being developed and acquired (see [5, 6, 7, 8] for detailed how-to suggestions).

MITRE Value Impact Assessment: Collaborative Tool to Use with POET, TEAPOT, and SE Profiler

Value metrics charts were developed in 2004 to portray MITRE's range of relationships with a particular customer and the scope and nature of MITRE's work for that customer [9]. Two main types of value metrics have been developed to: (1) address criticality of the mission need vs. the nature of MITRE's work (i.e., highly repeatable vs. advancing the state of the art); and (2) address MITRE's relationship with a customer compared to the scope of our work for them. Value metrics charts can be generated from inputs prepared in Excel.

The primary goal for using the MITRE Value Impact Assessment is to strengthen work program content, customer relationships and satisfaction, and MITRE's impact. This tool is often used to identify future directions for MITRE's engagement model and differentiation with a customer (e.g., projecting MITRE to take on a more strategic role, or in some circumstances, transferring a repeatable role to a government contractor to maintain).

Stakeholder Analysis Process: Collaborative Tool to Use with POET, TEAPOT, and SE Profiler

The stakeholder analysis process is used to strengthen relationships among key stakeholders by establishing why different stakeholder types behave differently and why they behave the way they do. Stakeholder analysis enables tailoring strategies for key stakeholders to take greater advantage of opportunities and avoid or mitigate unwanted risks when they become apparent.

Though the direct customer relationship is a high priority, it is important to determine which other stakeholder types are of a priority and undertake relationship improvement efforts with them.

Once the key stakeholders have been established, a relationship management program starts by developing a relationship management plan. The tips for the customer relationship can be adapted in planning, executing, and assessing a relationship management program with other key stakeholder types.

Enterprise Principles: Collaborative Tool to Use with POET, TEAPOT, and SE Profiler

Enterprise principles are enduring guidelines that describe the way an organization fulfills its mission. Principles express an organization's intentions and fundamental values so that decisions can be made from a common understanding.

Principles are driven by functional capability and/or organizational visions, strategic plans, enterprise direction, and policy directives, which in turn are generally driven by presidential executive orders, legislation, and other external mandates and directives (see [10] for additional details).

The primary intended audience for enterprise principles includes mission capability proponents, chief information officers, chief architects, and program managers.

Models for Enterprise Federation Analysis: Collaborative Tool Used with POET, TEAPOT, and SE Profiler

Federal Enterprise Architecture

While the federal government is organized into agencies, departments, and other organizational structures, many of the government's functional missions cross agency boundaries and authorities. To address the need to coordinate efforts and plans across federal agencies and to share information and services, the Office of Management and Budget (OMB) has established the Federal Enterprise Architecture (FEA). The structure of the FEA is maintained by OMB, but portions of it, called segments, are developed and maintained by agency leads in coordination with other agencies. Cross-agency FEA segments are documented by OMB in the Federal Transition Framework [11], which is used in life-cycle planning activities of agencies and their budget submissions. Agencies are responsible for submitting segment architectures to OMB. A federal segment architecture methodology was developed to provide guidance and direction to agencies for developing their segment architectures; it consists of a collection of best practices, tools, techniques, templates, and examples of the various elements that may be included in a segment architecture [12].

Table 1. Summary of Analysis and Collaborative Tools

Analysis Tools	Topics/Areas to Address and Analyze
POET TEAPOT SE Profiler	Scope of work program/project Work program/project relevant to customer's mission and strategic objectives MITRE roles and responsibilities Work program/project environments (political, operational, economic, and technical) Relationships with the customer Work program/project management (planning, implementing, and monitoring) Work program status (accomplishments, actions, and timeliness)
Collaborative Tools	**Topics/Areas to Address and Analyze**
MITRE Value Impact Assessment	Work program/project relevant to customer's mission and strategic objectives MITRE roles, responsibilities, and impacts
Stakeholder Analysis	Work program/project environments MITRE internal stakeholders Customer stakeholders Program stakeholders Relationships with stakeholders
Enterprise Principles	Work program/project compliance to customer's enterprise objectives Work program/project environments (e.g., standards, integration, sharing, etc.) Work program/project management (initiation, planning, execution, and closing)
Models for Enterprise Federation Analysis	Work program/project compliance to mandates and policies

Department of Defense (DoD) as a Federated Enterprise

The DoD, like many agencies, has missions to perform that cut across its organizational elements. In addition, there are common business functions, such as financial, management, and IT infrastructure needs that cut across both missions and organizational elements. To address the many potential relationships, and ultimately both complementary and competing interests, the DoD has been developing and employing a federated enterprise approach to provide consistent context and disciplines for accomplishing the mission of the Department collectively [13, 14, 15], as are other federal agencies [16, 17, 18, 19, 20].

References and Resources

1. “Stakeholder Analysis and Relationships,” MITRE Project Leader Handbook, The MITRE Corporation.
2. MITRE Systems Engineering (SE) Competency Model, September 1, 2007, Ver. 1, The MITRE Corporation,
3. The MITRE Corporation Center for Enterprise Modernization, February 10, 2009, Quality Handbook, Ver. 2.0, pp. 4–33.
4. The MITRE Corporation “TEAPOT Chart: Characterize Systems Engineering Output.”
5. Carlock, P. G., S. C. Decker, and R. E. Fenton, Spring/Summer 1999, “Agency-Level Systems Engineering for ‘Systems of Systems,’” *Systems and Information Technology Review Journal*, pp. 99–110.
6. Stevens, R., July 2010, *Engineering Mega-Systems: The Challenge of Systems Engineering in the Information Age*, CRC Press.
7. Stevens, R., “Profiling Complex Systems,” The MITRE Corporation.
8. Stevens, R., “Managing Uncertainty,” The MITRE Corporation.
9. The MITRE Corporation, “Use Value Metrics to Assess Potential Technical Work.”
10. The Open Group Architecture Framework (TOGAF), “Architecture Principles,” TOGAF version 8.1.1, Pt. IV, Resource Base, Ch. 29.
11. Federal Transition Framework (FTF), Ver. 2.0, www.whitehouse.gov, January 2008,
12. Office of Management and Budget, Federal Enterprise Architecture (FEA), www.white-house.gov.
13. DoD Deputy Chief Information Officer, “DoD Architecture Framework 2.0, Architecture Development, Enterprise Architecture,” http://dodcio.defense.gov/dodaf20.aspx, retrieved July 29, 2010.
14. Golombek, A., and W. Okon, “EA Federation and Building the DoD EA—Briefing to OMG,” DoD CIO, September 16, 2009. UPDM is an Object Management Group (OMG) initiative to develop a modeling standard that supports both the DoDAF and the UK Ministry of Defence Architecture Framework (MODAF). The modeling standard is called the Unified Profile for DoDAF and MODAF (UPDM).

15. DoD CIO, “DoD Governance: Architecture Federation,” July 29, 2010 (requires Intelink username and password).
16. The MITRE Corporation, December 3, 2003, United States Coast Guard Enterprise Architecture Framework, Ver. 0.3
17. Federal Health Architecture, www.healthit.gov.
18. Mullins, K., December 15, 2005, DOJ Litigation Case Management (LCM) Target LCM Architecture.
19. Department of Defense Office of Chief Information Officer, May 2009, Defense Information Enterprise Architecture, Ver.1.1.
20. Grasso, D., and M. B. Burkins, December 1, 2009, “Holistic Engineering Education Beyond Technology,” Springer, Ch. 5.

Enterprise Planning and Management

Definition: *Enterprise planning and management addresses agency and program direction, monitoring, and resource allocation to achieve goals and objectives at the strategic level.*

Keywords: *governance, performance management, portfolio management, program management, resource allocation, strategic planning*

Introduction

Enterprise planning and management takes a strategic view of the major plans and processes needed for a federal government organization to achieve its mission. The legislative branch does not often get into details about which components of an executive branch agency will execute each aspect of the mission, or how they will operate. Therefore, at the strategic level, each agency must plan, manage, and account for both how and to what extent it achieves that mission. MITRE engineers may be asked by sponsors to help develop and execute these strategic-level plans and processes.

An awareness and working knowledge of enterprise planning and management is needed by all systems engineers whether or not their daily activities directly support enterprise-level government activities. Nearly all government development programs or those undergoing significant modifications are already interfacing to a number of other systems, networks, databases, and data sources over the Web, or are part of a family or system of systems. Therefore, at whatever level of the enterprise MITRE systems engineers operate, enterprise planning and management provides the context for or environment in which they execute their activities.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to bring an enterprise perspective to their support of customer planning and management activities at whatever scale of the enterprise they operate: subsystem, system, system of systems, or enterprise.

When directly supporting enterprise planning and management activities, MITRE systems engineers are expected to understand the central role systems engineering plays in effectively planning and managing the evolution or modernization of government enterprises. MITRE SEs are expected to tailor and adapt systems engineering principles, processes, and concepts to match the scope and complexity of the government overall effort as well as the agency or department acquisition regulations, policies, and governance approaches. MITRE SEs need to be cognizant of enterprise management challenges/issues so they can assume appropriate accountability for the success of the activities they support. MITRE staff are expected to coordinate extensively across the corporation, other FFRDCs, academia, and industry to ensure that all aspects of the problem are considered in shaping products or decisions. MITRE contributions should provide an enterprise perspective, be product and vendor neutral, and anticipate future missions and technologies.

Best Practices and Lessons Learned

Do the right things and do them well. "The greatest waste in business is doing the wrong thing well"—Henry Ford.

The primary objective of enterprise planning and management is to ensure that the enterprise is doing the right things—directing its resources with maximum impact for achieving its mission. Doing the right things well is more a tactical conern, with program and project execution. The best practices and lessons learned apply to planning to do the right things, and then monitoring how (and whether) doing those things is leading toward the end goals.

Provide the right focus. Focus organizational resources on achieving the goals outlined in the strategic plan.

Importance of senior leadership role. An essential component of success is the commitment and active involvement of the organization's senior leadership.

The articles under this topic provide more detailed descriptions of best practices.

Articles Under This Topic

The “IT Governance” article outlines government enterprise investment management policies and goals and describes best practices for governing those investments in the federal government.

The “Portfolio Management” article describes how MITRE provides technical advice and recommendations to support the customer in making resource allocation decisions to achieve desired outcomes within funding and other business constraints.

The “How to Develop a Measurement Capability” article describes the general principles and best practices of performance measurement methods and systems and how to use performance measures to assess progress toward achieving strategic goals and objectives and to inform decisions about resource allocation.

References and Resources

1. U.S. Government Accounting [now Accountability] Office (GAO)/General Government Division, May 1997, Agencies’ Strategic Plans Under GPRA: Key Questions to Facilitate Congressional Review, Ver. 1, GAO/GGD-l0.l.16.
2. GAO, Information Technology Investment Management: A Framework for Assessing and Improving Process Maturity, 2004 (GAO-04-394G).
3. GAO, Government Performance: GPRA Modernization Act Provides Opportunities to Help Address Fiscal, Performance, and Management Challenges, March 2011 (GAO-11-466T).

Definition: *Governance is the responsibilities, structures, and processes by which organizations are directed and controlled [1]. It defines how business, engineering, and operations decisions are made to support business strategy. Enterprises have many interrelated layers of governance that differ in scope and decisions. Governance definitions vary, but they have some elements in common:*

It is about making decisions to support business strategy. It requires a framework that defines roles and responsibilities, processes, policies, and criteria for sound decision making. It requires identifying the right people to make and be held accountable for tough decisions.

Keywords: *business process, framework, governance, strategy*

ENTERPRISE PLANNING AND MANAGEMENT

IT Governance

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand why IT governance is a critical issue for the federal government and the integral role IT governance serves within organizational strategic planning. They are expected to assist the customer in adhering to the requirements of the organization's governance program, establishing appropriate roles and responsibilities, and following mandates and best practices for governing IT investments in the federal government. MITRE SEs also should play a role in helping an organization achieve real value from IT investments by ensuring alignment to the enterprise strategies and governance program. MITRE SEs' role is to increase the value of the IT investments by providing feedback and lessons learned on how the governance program

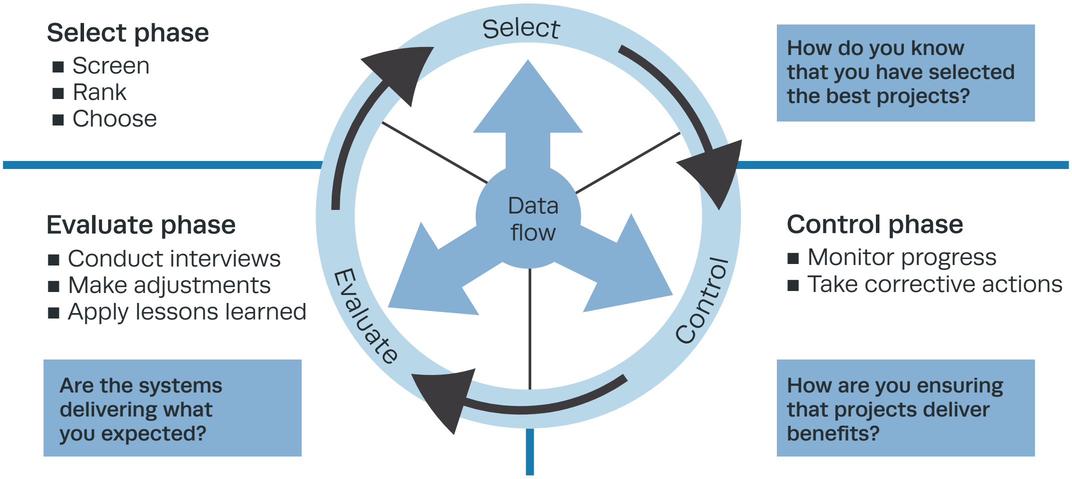

Select phase
■ Screen
■ Rank
■ Choose
Select
How do you know that you have selected the best projects?
Evaluate phase
■ Conduct interviews
■ Make adjustments
■ Apply lessons learned
Data flow
Control phase
■ Monitor progress
■ Take corrective actions
Evaluate
Control
Are the systems delivering what you expected?
How are you ensuring that projects deliver benefits?

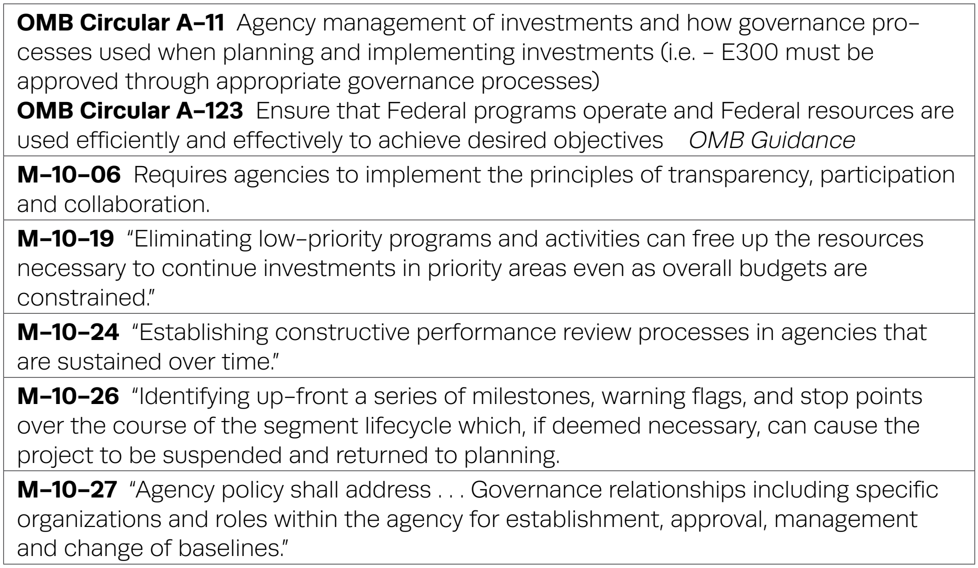
Figure 1. Fundamental Phases of the IT Investment Approach

is functioning and where improvements should be made. MITRE SEs are expected to establish a foundation on which good decisions can be made by deriving and analyzing data for specific decisions (e.g., those related to business cases, reference architectures, policies, standards, formats, processes, and life cycles needed to establish governance). This may require an understanding of organizational change and transformation, risk management, and communications planning. For more information on both of those topics, see the SEG topic Transformation Planning and Organizational Change.

Background

Enterprise governance is a set of responsibilities and practices exercised by "a board and executive management with the goal of providing strategic direction, ensuring that objectives are achieved, ascertaining that risks are managed appropriately, and verifying that the enterprise's resources are used responsibly [2]." All other types of governance within an organization—IT governance, interagency governance, program governance, project governance—are within the context of enterprise governance.

Information technology (IT) governance is an integral part of enterprise governance and consists of the leadership, structures, and processes that ensure that an organization's IT sustains and extends its strategies and objectives [2]. IT governance requires a structure and processes to support repeatable decision making, alignment of IT activities to the enterprise's strategic goals and objectives, and a clear understanding of authority and accountability.



OMB Circular A-11 Agency management of investments and how governance processes used when planning and implementing investments (i.e. – E300 must be approved through appropriate governance processes) **OMB Circular A-123** Ensure that Federal programs operate and Federal resources are used efficiently and effectively to achieve desired objectives *OMB Guidance*
M-10-06 Requires agencies to implement the principles of transparency, participation and collaboration.
M-10-19 "Eliminating low-priority programs and activities can free up the resources necessary to continue investments in priority areas even as overall budgets are constrained."
M-10-24 "Establishing constructive performance review processes in agencies that are sustained over time."
M-10-26 "Identifying up-front a series of milestones, warning flags, and stop points over the course of the segment lifecycle which, if deemed necessary, can cause the project to be suspended and returned to planning.
M-10-27 "Agency policy shall address . . . Governance relationships including specific organizations and roles within the agency for establishment, approval, management and change of baselines."

Figure 2. Sample OMB Circulars and Guidance

As with any governance body within an organization, IT governance cannot be viewed, assessed, modified, or changed without considering the rest of the organization's governance bodies and practices.

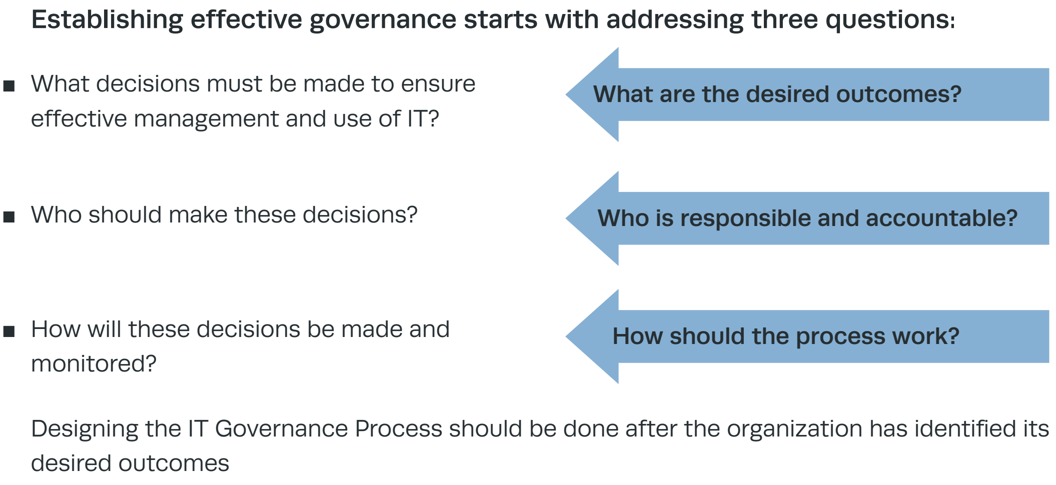
IT governance affects the degree to which an organization will get value from its IT investments. The goals of IT governance are to ensure IT investments generate business value and to mitigate IT risks [6]. Research among private sector organizations has found that "top performing enterprises succeed in obtaining value from IT where others fail, in part, by implementing effective IT governance to support their strategies and institutionalize good practices [3]." This principle can be extended to the goals of the enterprise at large. Whereas the purpose of enterprise governance is to effectively derive value from the enterprise resources for all the constituents in the enterprise, based on defined enterprise goals and strategy, the purpose of IT governance is to ensure the effective and efficient management and delivery of goods and services aligned to enterprise strategies [6]. For more information on Enterprise Strategy, see the article in this section on Strategic Planning. Also, see related articles under the Enterprise Technology, Information, and Infrastructure topic in this section.

For nearly two decades, the federal government has been trying to adopt investment and usage best practices from private industry to ensure that IT enables government to better serve the American people. Through legislation, executive orders, and guidance, the federal government requires that agencies apply rigor and structure to the selection and management of IT in order to achieve program benefits and meet agency goals. In 1996, Congress passed the Clinger-Cohen Act, which required, among other things, that senior government decision makers become involved in the decisions concerning the value and use of IT in the organization.

IT Investment Management

In 2004, the U.S. Government Accountability Office (GAO) published Information Technology Investment Management (ITIM): A Framework for Assessing and Improving Process Maturity [4]. ITIM is a maturity model built around the select/control/evaluate approach outlined in the Clinger-Cohen Act. ITIM establishes requirements for IT management and is used to assess how well an agency is selecting and managing its IT resources (Figure 1). In many agencies today, the IT investment life cycle includes a fourth phase: Pre-Select. In this phase, the organization plans and evaluates its strategy and mission needs before the select phase and "pre-selects" those that best help the organization meet this strategy before final selection of investments.

The Office of Management and Budget (OMB) has issued executive orders and circulars to help improve agency management of IT resources to support and govern the intent of Clinger-Cohen and other legislation. (See Figure 2.) These circulars approach the problem of the use of IT through the budget process requiring that requests for funds for IT investments meet specific requirements.



Establishing effective governance starts with addressing three questions:

- What decisions must be made to ensure effective management and use of IT?
- Who should make these decisions?
- How will these decisions be made and monitored?

What are the desired outcomes?
Who is responsible and accountable?
How should the process work?

Designing the IT Governance Process should be done after the organization has identified its desired outcomes

Figure 3. Establishing Effective Governance

Most recently, OMB issued its 25 Point Implementation Plan to Reform Federal IT Management [5], once again addressing the concerns around federal agencies' ability to achieve effectiveness and efficiency from its IT investments. The 25 Points institutes a "TechStat" process within the federal government to take a close look at poor or underperforming programs.

Best Practices and Lessons Learned

The governance program must have clear goals and defined outcomes tied to strategic goals. One of the first actions in standing up a governance program is to clearly define and articulate the scope of what is being governed and the desired outcomes of governance decision making. The outcome of the governance process should be aligned to the organization's strategic goals and clearly communicated to all stakeholders in the organization. The focus on outcomes will drive all other decisions surrounding the establishment of the governance program, including what decisions need to be made, who should make the decisions, and what data and analysis are needed. (See Figure 3.)

Often, an organization does not articulate the real objectives of the governance program, or the governance efforts are focused solely on complying with federal laws and guidance. It is not uncommon for an organization to spend considerable resources developing charters, processes, and governance structures without a clear and universal understanding of the goal. And, although compliance is certainly important, if it is the only focus of the program, it is not likely provide real value to the organization. Accepting a broader view on the need for governance, an IT governance body could have goals focused on value delivery, resource management, and/or risk management where compliance objectives are simply part of overall decision making.

Ensure reliable information for decision making. Successful and effective governance relies on the availability of reliable data and information on which to base decisions. Organizations often do not have the right information about projects or investments to make good decisions. The result can be "garbage in, garbage out." Once an organization has defined its desired outcomes for the process, it can begin to identify the information needed to make decisions to achieve these outcomes. This type of information would include, for example, a project's actual cost, schedule, and scope performance against the estimated or projected performance. IT management documentation, service management monitoring, and configuration management also inform the decision-making process. Data for IT decision making includes assessment factors such as return on investment, total cost of ownership, performance measurements, IT security, and enterprise architecture; development of scoring algorithms; and guidelines and methodology, as required, for consistency in scoring. SEs can assist by investigating alternative courses of action, determining the applicable measures of effectiveness, and relating these to assessments of risk (including technical maturity and applicability to the task at hand), cost, schedule, and performance. If the information is not readily available, executive sponsors can help support a process for getting the right information to decision makers in a predictable manner.

Governance programs must gain and retain the executive sponsorship needed. Lack of leadership for establishing and maintaining a governance program is a challenge to sustaining it over time. A related issue is changing leadership. Often a federal executive establishes and puts full weight behind a program, only to leave behind a successor who does not support the cause as vigorously. This underscores the need for a sustained, documented, and formalized program focused on clear IT outcomes aligned to organizational strategy. The program needs to provide opportunities to revisit it for updates and to ensure that team members and stakeholders are sufficiently engaged.

Governance requires a structure, defined and repeatable processes, policy, and criteria. Once the desired outcomes of governance are identified, an organization needs to establish the decision-making authority and the participants' roles and responsibilities. This involves the development of a governance structure that establishes the authority of governance bodies, processes that establish repeatable criteria and decision making, and preparation of charters, or similar type of documents, to describe the scope, duties, structure, and selection process of members, roles, and responsibilities. For governance to be effective over a sustained period of time, it is more likely to succeed if it reflects the culture and decision-making style of the organization and is integrated with existing decision making, tolerance of risk, and operational management processes. The governance processes should not be burdensome, but can and should be tailored and developed to ensure a "fit to purpose" by matching the size and scope of the program/organization business needs and strategic goals to the climate, risk tolerance acceptance levels, and governance maturity level of the organization.

Performance measures are critical to effective IT governance. Many organizations find it difficult to measure the performance of their IT governance programs because the programs often don't function in the context of governance goals but instead focus on individual IT project goals. In these situations, the lack of effective governance measurements limits the understanding of how well the process is performing in meeting the decision-making needs of the organization. Successful governance activities maintain reporting or tracking of measures that indicate the value of the governance program for its intended purpose toward meeting defined goals. Examples of IT governance performance measures focused on improving the process include increasing transparency of IT investment decisions, demonstrating an increase in IT innovation investments with a decrease in IT sustainment spending, and incorporating flexibility in IT infrastructure to react to changes in regulation and policy environment [7]. Regular reporting not only serves to show value, but also helps maintain the focus of the governance program as it executes. MITRE SEs can help customers measure and report on performance indicators to enable governance bodies to make decisions about projects and programs in the context of the organization's goals.

Articulate the value of governance to balance its perception as a burden. Because organizations often have the notion that governance is too burdensome, in order to meet release or development schedules, their governance processes are often short-cut or bypassed altogether. This may appear to provide short-term rewards, but experience has shown it is inefficient in the long term. As organizations try to balance resources across many efforts, their visibility into the programs diminishes and, as result, they lose opportunities for consolidation or more effective enterprise operations that would have been achieved if they had had a functioning governance process.

Summary

To be successful, IT governance must be integrated and aligned with the organization's enterprise governance. The decisions for IT investments must have a direct connection to supporting goals defined by the organization and to the allocation of resources to meet those goals. IT governance decisions should have a clear line of sight to the agency's goals and intended strategic outcomes. IT governance activities provide focus and create a path forward to meeting the information management challenges faced by the agency.

There are many approaches to implementing effective governance. The exact approach depends on the strategy and results the organization is trying to achieve as well as the culture within which the organization operates. A review of governance practices suggests that specific foundational elements must be in place for governance to be effective:

- Strong executive sponsorship of the process
- Clear and well-communicated strategic goals
- Clear, well-defined roles and responsibilities
- Standardized data and information transparency
- Measurement and planned review of the governance practices to ensure value

Governance frameworks that may be of interest include CoBIT, ITIL, CMMI, and ISO38500.

References and Resources

1. International Standard ISO/IEC 38500:2008(E), 1st Ed., 2008-06-01.
2. ITGI Board Briefing on IT Governance, 2nd Ed.
3. Weill, P., "Don't Just Lead, Govern: How Top Performing Firms Govern IT," Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology, 2004.
4. GAO Executive Guide, Information Technology Investment Management: A Framework for Assessing and Improving Process Maturity, March 2004. GAO-04-394G.

5. Office of Management and Budget, 25 Point Implementation Plan to Reform Federal Information Technology Management, December 9, 2010.
6. R. Brisebois, G. Boyd, and Z. Shadid, "Canada - What is IT Governance? And Why is it Important for the IS auditor," *The IntoSAI IT Journal*, No. 25, pp. 30–35, August 2007.
7. Fink, K., and Ploder, C. Decision support framework for the implementation of IT-governance. Hawaii International Conference on System Sciences, pp. 432–441, January 2008.

Additional References and Resources

Weill, P., and J. W. Ross, *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business Press, 2004.

Definition: *A continuous and persistent process that enables decision makers to strategically and operationally manage resources to maximize accomplishment of desired outcomes (e.g., mission results, organizational improvements, enhancement of operational capabilities) within given constraints and constructs such as regulations, interdependent architectures, budgets, concept of operations, technology, and mission threads.*

Keywords: *capability, capabilities, optimize, outcomes, portfolio analysis, portfolio management, portfolios*

ENTERPRISE PLANNING AND MANAGEMENT

Portfolio Management

MITRE SE Roles and Expectations: MITRE systems engineers are expected to understand and keep abreast of sponsor and customer portfolio management (PfM) challenges, themes, and strategies. They are expected to recommend and apply systems engineering approaches to address PfM opportunities and issues, including data-driven analysis, incremental baseline innovation, time-certain and price-driven agile acquisition, and the exploitation of commercial development methods and products.

Overview of Portfolio Management

Modern portfolio theory provides foundational concepts that are useful in multiple portfolio management environments. Portfolio management is about aggregating sets of user needs into a portfolio and weighing numerous elements to determine the mix of resource investments expected to result in improved end-user capabilities. The key elements that portfolio management must assess are overall goals, timing, tolerance for risk, cost/price, interdependencies, budget, and change in the enterprise environment over time.

Accountability for and transparency of government expenditures has been a significant focus during the last two decades. More recently, it has become important that these expenditures address key enterprise (agency, mission) outcomes efficiently, effectively, and collectively rather than as independent and unrelated initiatives. Portfolio management is a key tool for supporting this form of fiscal accountability. A simplified overview of portfolio management activities is provided in Figure 1. Various laws, directives, and guides relate to portfolio management.

At present there are two major, definitive types of portfolio management: (1) information technology (IT) portfolio management and (2) capability portfolio management (CPM). IT portfolio management deals with investment analysis from a hardware and software perspective for an enterprise: dealing with the configurations and evolution for IT assets, re-capitalization, savings through concepts like regionalization, virtualization, shared assets, cloud capabilities, etc. CPM deals with managing the end user capabilities (applications, data, services) as investment options and selecting the best set of functional capabilities to invest resources in and to evolve over time. Government organizations are currently in various stages of implementation with multiple approaches, and they have met with different levels of success. MITRE systems engineers can use our knowledge to help analyze the best way forward for successful customer IT architectures and implementations, as well as use our knowledge of the operational needs and associated capabilities within the enterprises to help

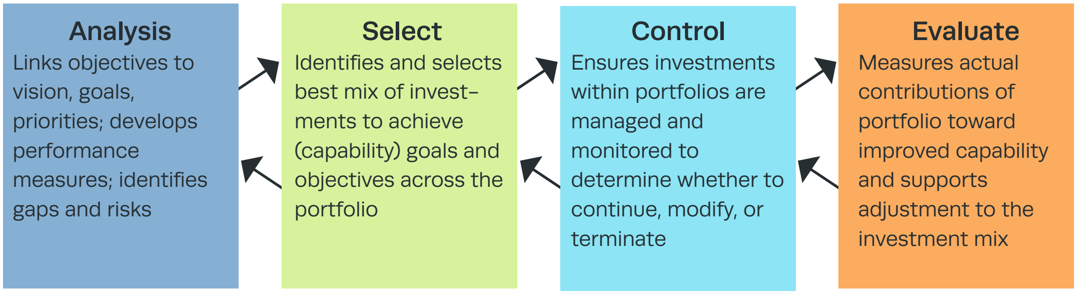

Analysis
Links objectives to vision, goals, priorities; develops performance measures; identifies gaps and risks
Select
Identifies and selects best mix of invest-ments to achieve (capability) goals and objectives across the portfolio
Control
Ensures investments within portfolios are managed and monitored to determine whether to continue, modify, or terminate
Evaluate
Measures actual contributions of portfolio toward improved capability and supports adjustment to the investment mix

Figure 1. Simplified View of Portfolio Management Activities

customers performing CPM. There is a place for both types of portfolio assessments. As an enterprise conducts CPM, they will undoubtedly need to construct the best IT environment to support the capabilities.

Stakeholder Engagement, Roles, and Responsibilities

Portfolio management requires leadership commitment. Leadership must endorse portfolio management goals, a rigorous and analytical process, and the willingness to make difficult recommendations and decisions such as investment termination. MITRE can help with leadership commitment by analyzing the options and formulating courses of action that define the best investment and use of resources and by highlighting the cost-benefit and return on investment from the recommended application of resources.

Engage all stakeholders early and often. Due to the significant number of portfolio capability providers, as well as the organizational constructs/governance structures that may divide the decision maker and the portfolio managers, it is important to identify all stakeholders and to understand the magnitude of their stake in the portfolio and how specific stakeholder groups might drive portfolio components and the portfolio. Understanding the different roles, responsibilities, and perspectives of the stakeholders (including those of your particular customer) helps in devising strategies to ensure objective assessment of potential investments, stakeholder buy-in, viable and affordable recommendations, and minimization of “back-door” efforts. Knowing how each stakeholder group drives the portfolio can suggest the needed level of attention that must be paid to each. A minority stakeholder may drive a single requirement that drives solutions to be significantly more complex and costly than would a majority stakeholder holding 90 percent of the requirements—a situation that the government must avoid.

Recommended Tools and Techniques to Use in Portfolio Management

Using the process of Figure 1, the following sections describe the tools and techniques along with the actions of systems engineers to help accomplish portfolio management.

Analysis Tools and Techniques

Establish the common set of operational needs over time. Understand what requirements, capabilities, goals, or outcomes need to be achieved, when they must be delivered, how they are measured, and how they are prioritized. This information provides specific and common targets for each element of the portfolio. However, the collection of this data is not normally standardized and sustained in a meaningful way in all organizations. The development and maintenance of this information should be a goal of the collective stakeholders of the portfolio. The SEG topics Concept Development and Requirements Engineering provide articles on how to help manage the concepts, needs, and requirements of users and can clarify the priorities of these as input to portfolio management.

Establish an analytic process. The government needs to move away from using forcefully conveyed arguments that appeal to group emotions to using an analytical foundation for portfolio decisions. To build a compelling case for portfolio selection (and continuation), help establish a well-understood and transparent analytic process with a framework that addresses specified portfolio goals/outcomes, incorporates key contextual elements (e.g., scenarios) and criteria/metrics important to the decision, uses appropriate data sets, and allows consistent comparison across the portfolio options. Execution of this process should provide insight into the trade-offs and define how the portfolio will be chosen or modified. Figure 2 provides an example of what is called an "efficient frontier" created using MITRE's Portfolio Analysis Machine (PALMA™) optimization tool. The efficient frontier showcases the portfolio options (black and red points) that provide the most "bang for the buck" (defined here by an overall benefit measure for a given budget). It also shows gray points that are a subset of the less efficient portfolios. MITRE systems engineers should understand the mechanics and value of using tools such as PALMA to better understand trade-offs and should call on MITRE experts such as those in MITRE's Center for Acquisition and Systems Analysis and other analysis organizations like the National Security Analysis Group to help with the analyses.

Be data driven. Organizations generally do not have a repository that provides a single, consistent, and robust database to support organizational portfolio management. Based on time and availability, the team (including MITRE systems engineers) should develop the best data set to assess criteria and create consistent comparisons across the investments considered. Though the most objective data is sought out, the best data available may actually come from subject matter experts (SMEs). MITRE systems engineers can help facilitate cost-benefit assessments with groups of SMEs by providing questionnaires and facilitated discussions of proposed capability benefits, prioritization, process improvements, resource savings, etc.

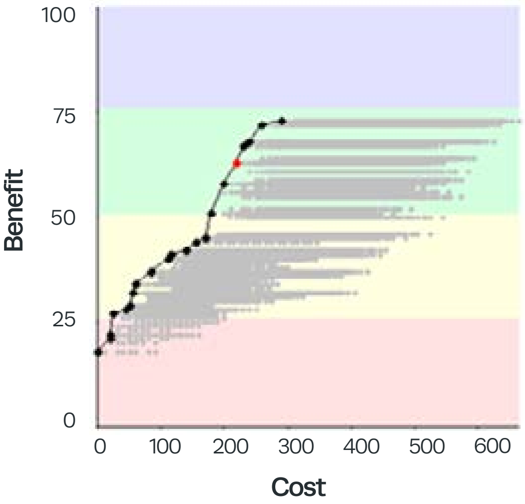


Figure 2. Example Efficient Frontier

Understand the contents of a portfolio. A full understanding of the investments in a portfolio, as well as of those that may be related to or impacted by your portfolio, is required in order to define the correct trade-offs for decision making. A common understanding of the portfolio content by the decision makers, reviewing bodies, stakeholders, and systems engineers is critical. MITRE systems engineers can help lay out the contents of the portfolio, their interconnections, feasibility, risks, etc. Modeling and simulation and in some cases actual prototyping and experimentation can be used by MITRE systems engineers to highlight the features of the critical drivers in the portfolio.

Determine dependencies. Current operations require multiple systems, system components, etc., to work together to create appropriate capabilities, which in turn creates dependencies within portfolios that must be identified and understood. The SEG Risk Management topic provides some guidelines on dependency risks. Multiple data sources, artifacts, policies, etc., must be considered in total, as well as an understanding of the links across these elements, to gain full comprehension of the complexity of the portfolio. For example, we need to ensure that we understand the connection between elements such as requirements, capabilities, mission threads, architectures, and systems.

Selection Tools and Techniques

Know the baseline. Based on an understanding of user needs, the team must understand how current needs are being met before recommending changes to the portfolio. In some cases this is called development of the "baseline." MITRE systems engineers help establish the baseline using techniques like federated architectures (see the SEG article, "Architectures Federation") where the baseline and subsequent evolution of the proposed portfolio can be captured.

Adequacy of the options. A robust set of options allows key insights into the trade-offs and their drivers to address portfolio offset drills and changes in funds. Various levels of the options may be addressed, including alternate acquisition strategies, different risk profiles, and different cost-effectiveness profiles. MITRE SEs can help the customers understand the pros and cons of each option, including feasibility, risk, performance, cost, and schedule considerations.

Control Tools and Techniques

It's more than technology. The business and programmatic aspects (including cost, acquisition strategies, business models, and risk) of the entire portfolio and its components are as important as the technical aspects.

How to buy is as important as what to buy. Defining options within the portfolio should include consideration of how the option should be acquired with consideration of timing and cost. MITRE systems engineers typically help acquisition organizations with strategies and methods and help them extend and apply this knowledge at the enterprise and its portfolio level.

Establish an integrated master schedule (IMS). An IMS with a life-cycle view is essential to reflect key technical and programmatic interdependencies of portfolio components and allows for focus on synchronization and integration points. System integration analyses can be performed by MITRE systems engineers to determine the schedule dependencies, impacts of slippages, ability to synchronize with other efforts, and assessment of when the portfolio needs to be re-assessed based upon schedule challenges. See the articles "Identify and Assess Integration and Interoperability (I&I) Challenges" and "Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies" for insights on systems integration.

Evaluation Tools and Techniques

Establish outcomes for the portfolio and appropriate metrics to monitor progress. Metrics are difficult to establish, in part because they must reflect common recognition of outcomes across the portfolio. But they are critical to measuring and tracking efficiency and effectiveness. MITRE systems engineers can help formulate metrics through knowledge of the enterprise's operational concepts, needs, requirements, mission accomplishment and assurance, and the operational and technical trade-offs for these needs.

What's the value proposition? Each investment, program, or resource used must determine its mission as well as how it supports the outcomes/products of the portfolio in which it resides. The cost and funding profile, effectiveness, timeliness of delivery, and risks of each component in relation to the portfolio must be understood. MITRE systems engineers usually focus on the results of getting capabilities to the end users to conduct this mission. Having this knowledge and emphasis and encouraging this perspective across the portfolio stakeholders will help keep the emphasis on the users' value.

Issues and Challenges Impacting Successful Portfolio Management

MITRE system engineers should understand the big picture when it comes to portfolio management and ensure that appropriate perspectives, information, analysis, and tools are brought to bear on the issues.

MITRE systems engineers should understand where the system, system of systems, enterprise, and organization they support fits in the relevant portfolio(s); how it impacts or is impacted by the portfolio investment decisions and its elements; what overarching outcomes need to be supported/achieved and why; and the statutory, regulatory, and policy environment affecting the portfolio decisions for the enterprise. MITRE systems engineers should ensure that appropriate analytics, tools, data sets, and strategies are brought to bear and that appropriate consideration is paid to stakeholders.

The issues related to portfolio management include:

Dueling Directives. There may be multiple directives within portfolio management that must be understood in the context of your program or portfolio. These directives may have come from various governing organizations and have conflicting and inconsistent guidance that is difficult to apply to the portfolio assessment. Knowing how your program, portfolio, and/or organization fits into one or more of these structures should help identify these disconnects and support your work to ensure progress and appropriate accountability. MITRE systems engineers can help highlight the inconsistencies and work with the responsible organizations to provide clear and consistent guidance and governance to the whole enterprise.

Multiple Taxonomies. For many government organizations/agencies, there may be multiple taxonomies that define the portfolio structure. Typically, a single taxonomy has not been adopted, nor has an approach been developed to allow the taxonomies to be used together effectively to support the goals of portfolio management. Given this, the MITRE systems engineer may need to map across multiple taxonomies to correlate equivalent or similar perspectives/areas of interest.

Budget Authority. Budget authority may not rest with the portfolio manager, making the portfolio manager more of an advisory resource than a decision maker. The Clinger-Cohen Act suggests IT budget authority rests with the Secretary, the CFO, and the CIO of the particular federal department. In the Department of Defense, for example, budget authority generally resides with the Military Services (Title X) and not the capability portfolio managers. In cases where the portfolio manager also has budget authority, many times the execution of the investment plan can be streamlined. In cases where the responsibilities are in separate organizations, MITRE staff can help the portfolio managers create a persuasive case for the preferred portfolio and highlight the value/cost-benefits of applying the portfolio resources needed.

Budget Process. In many government agencies, budgets are planned for and executed at a lower level than a portfolio (e.g., program, program element, budget line, appropriation). This adds complexity, making portfolio management execution more difficult because the change recommendations may be at a more granular level than is reflected in budgetary accounting. MITRE systems engineers can maintain the investment profile using the detailed, data-driven analyses previously described and performing sensitivity analyses of changes to the individual components that comprise the portfolio.

Culture. Portfolio management is a “greater good,” or enterprise process, and is not supported within a program acquisition culture rewarded for individual program success rather than enterprise success. This is partly because it takes up-front investment to achieve a longer term “greater good” outcome. In addition, the mission success or portfolio savings benefits from portfolio changes are not adequately accounted for or attributed to the portfolio changes, making change a difficult proposition. MITRE systems engineers can demonstrate the greater good by presenting the value that the portfolio choice provides to the enterprise's concepts/needs, federated architecture, business planning, collective integration, design and development activities, and life-cycle cost.

Political Factors. Politics has consistently been an element of investment decision making, as have operational, economic, technical, and other factors. There may be cases where certain solutions are technically elegant and affordable but not politically feasible. It is important for MITRE to ensure consideration of all factors within the decision-space and to understand the implications of each. Substantive and validated analysis can illuminate whether an investment supports desired portfolio outcomes.

"Pet Rocks." In some cases, particular solutions may be favored by leadership. This may result in a less valuable investment being selected, which can undermine the ability to secure the most cost-effective portfolio. Analysis can and should inform an understanding of the "value proposition" of investments; however, it may be trumped by leadership preferences. MITRE SEs may recommend a solution that is not acted on for reasons outside of their control. When these situations arise, MITRE SEs should continue to highlight the risk and to provide an independent view of the situation while helping the government plan and execute their selected alternative.

Poor Life-Cycle Cost Analysis. Cost overruns are rampant in the government. This is partially due to the low levels of confidence inherent in the original cost estimates of the individual programs. Portfolio management further complicates these already "narrow" cost analyses by altering the foundational assumptions in the cost estimates. For example, a new innovation or a new approach to capability delivery may affect the development or sustainment costs of entire suites of investments. Integrating the effects of portfolio changes on the initial projected life-cycle cost estimate is confounded by flaws in the original cost estimates and by their inability to accommodate the PfM innovation. See the SEG article "Life-Cycle Cost Estimation" for practices on cost estimating that could be evaluated for use across portfolios.

References and Resources

Government Guidance, Policies, Regulations

40 U.S.C. 1401 et seq, Clinger-Cohen Act of 1996, Division E, National Defense Authorization Act for FY1996.

Air Force Instruction 33-141, Air Force Information Technology Portfolio Management and IT Investment Review, December 23, 2008.

CJCSI 8410.01, 22 Jun 07, Warfighting Mission Area Information Technology Portfolio Management and Net-Centric Data Sharing.

Department of Defense Chief Information Officer Desk Reference, August 2006, Foundation Documents, Vol. 1.

Department of Defense Directive 7045.20 USD(P), September 25, 2008, *Capability Portfolio Management*.

Department of Defense Directive 8115.01, October 10, 2005, *Information Technology Portfolio Management*, ASD(NII)/DoD CIO.

Department of Defense Directive 8115.02, October 10, 2005, *Information Technology Portfolio Management Implementation*, ASD(NII)/DoD CIO.

Federal CIO, Capital Planning, and IT Management Committee, Smart Practices Subcommittee, Performance Management Area, *A Summary of First Practices and Lessons Learned in IT Portfolio Management*, 2002.

GAO-04-394G, *Information Technology Investment Management, A Framework for Assessing and Improving Process Maturity,* March 2004, U.S. Government Accounting (now Accountability) Office, Executive Guide, Ver. 1.1.

GAO-07-388, *An Integrated Portfolio Management Approach to Weapon Systems Investment Could Improve DoD's Acquisition Outcomes*, March 2007, U.S. Government Accountability Office, Report to the Committee on Armed Services, U.S. Senate.

Health and Human Services Policy for IT – CPIC.

Net-Centric Functional Capabilities Board Endorsement of Capability Delivery Increments (CDI) Document, June 22, 2009, The Joint Staff, Memo.

OMB Circular A-11, August 2009, Preparation, Submission, and Execution of the Budget; Exhibit 53, Agency IT Investment Portfolio.

OSD Memorandum, 12 January 09 (predecessor: February 14, 2008), *Joint Capability Areas*.

Journal Articles, Technical Notes, White Papers

Enterprise Scale Portfolio Analysis at the National Oceanic and Atmospheric Administration (NOAA).

Markowitz, H., March 1952, "Portfolio Selection," *The Journal of Finance*, Vol. VII, No.1.

Articles, Books

Bhadra, D., and F. Morser, 26–28 September 2005, “Analysis of System-wide Investment in the National Airspace: A Portfolio Analytical Framework and an Example,” American Institute of Aeronautics and Astronautics, 5th ATIO/AIAA Conference.

The MITRE Corporation, Integrated Architecture-Based Portfolio Investment Strategies, technical paper.

Maizlish, B., and R. Handler, 2005, *IT Portfolio Management Step-By-Step*, John Wiley & Sons, Inc., Hoboken, NJ.

Moynihan R., Investment Analysis using the Portfolio Analysis Machine (PALMA) Tool.

Program Formulation and Project Planning,” MITRE Project Leadership Handbook, The MITRE Corporation.

Sanwal, A., April 2007, *Optimizing Corporate Portfolio Management: Aligning Investment Proposals with Organizational Strategy*, John Wiley & Sons, Inc., Hoboken, NJ.

Checklists, Toolkits

Corporate Executive Board, Portfolio Management.

DAU IT Portfolio Management Community.

ENTERPRISE PLANNING AND MANAGEMENT

How to Develop a Measurement Capability

Definition: *The GAO defines performance measurement as the ongoing monitoring and reporting of program accomplishments, particularly progress toward pre-established goals, typically conducted by program or agency management. They may address the type or level of program activities conducted (process), the direct products and services delivered (outputs), or the results of those products and services (outcomes). A program may be any activity, project, function, or policy with an identifiable purpose or set of objectives.*

Keywords: *evaluation, Government Performance and Results Act, logic model, measurement, outcome measures, outcomes, performance management, performance measurement, performance reference model, strategic planning*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the general principles and best practices of performance measurement methods and systems. They are expected to assist sponsors in developing a measurement capability in the systems acquisition and/or the operational organization. They assist in collecting and using performance measures to assess progress toward achieving strategic goals and objectives and to inform decisions about resource allocation.

Background

Congress required performance measures of all federal agencies starting in 1999. The legislation containing those requirements is the Government Performance and Results Act (GPRA), passed in 1993. The only federal legislation that requires strategic planning and performance measurement, GPRA requires each agency to develop a five-year strategic plan (to be updated at least every three years), an annual performance plan, and an annual performance report. In specifying what must be included in those documents, GPRA requires that a strategic plan must show the link between strategic goals and performance goals. A strategic plan must also contain an evaluation plan that includes performance measures.

GPRA designates the Office of Management and Budget (OMB) as the agency responsible for executing GPRA. OMB's process for evaluating agencies is called the Performance Assessment Rating Tool (PART). Two of the four sections of PART examine performance measures. Although PART is likely to change somewhat, the current Administration has announced that the fundamentals of that process will remain unchanged. Agencies must report performance, showing results (the "R" in GPRA). The Administration is increasing its emphasis on evaluation, which is a way to make sure that what matters is measured, that what is measured is really what is intended to be measured, and that results reported are credible.

Congress and the Administration are placing increased emphasis on performance and results for a simple reason: it is the best solution for achieving success when money is tight. Unless performance improves, there are basically three other, highly unpopular directions: (a) raise taxes, (b) cut programs, or (c) increase debt. MITRE can expect to see more requests for assistance with performance.

The Single Most Common Problem (and Its Solution)

At MITRE we are often asked to develop performance measures for a government program or other initiative. The most common problem about program performance cited in research—and that we have seen at MITRE—is that the program's goals/objectives have not been identified. *It is impossible to develop measures of progress if we do not know where we are trying to go.*

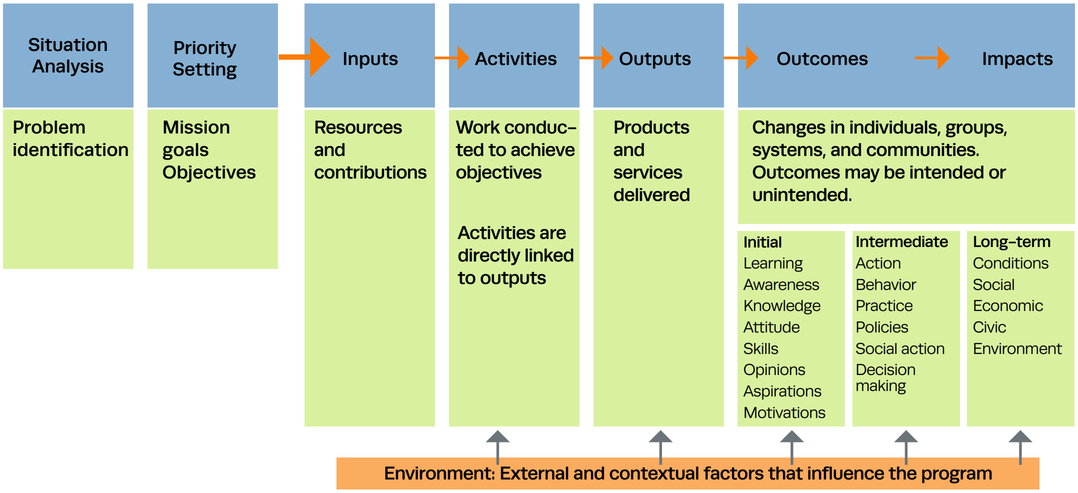
The first step in developing measures of progress is to identify the desired end point or goal. One of the most useful tools for identifying goals, and for developing performance measures, is the logic model. A logic model is a map, a one-page bridge between planning and performance.

The logic model shown in Figure 1 should be read from left to right.

- The problem that the program was created to solve is identified on the left.
- The agency's strategic priorities—its goals/objectives—are next and should directly relate to the problem.

- The next three columns are basic input-process-output. Inputs are people, funding, and other resources. Outputs are results of processes or activities. Output measures answer the question: "How do you know they really did that?" Outputs are usually expressed in numbers of units produced or units of service provided.

Outcomes are all about impact. They are answers to the question: "So what?" What difference did your product or service make? An initial outcome is softer, usually near-term, and might be measured by before/after tests of understanding if a training service were the output. Customer satisfaction is a common short-term outcome measure. An intermediate outcome might include changes in behavior, and it might be measured by finding out how many of those who received training are actually using their new skills. (*Note*: Often, short-term and intermediate outcomes are combined as intermediate outcomes.) Long-term outcomes are the conditions the program/agency is trying to change and should be a mirror image of the problem on the left of the logic model. Thus measures of long-term outcomes can be relatively easy to identify. A program established to address the problem of homelessness among veterans, for example, would have an outcome measure that looks at the number and percent of veterans who are homeless. (Defining "homeless" may be a separate issue to be addressed in a later implementation of data collection and reporting.)



Sources: GAO-02-923, Strategies for Assessing How Information Dissemination Contributes to Agency Goals. GAO/GGD-00-10, Managing for Results: Strengthening Regulatory Agencies' Performance Management, Ellen Taylor-Powell, 2000. A Logic Model: A Program Performance Framework, University of Wisconsin, Cooperative Extension Program Evaluation Conference.

Figure 1. Defining Performance Measures with Logic Models

Environmental factors can influence all stages of a program and need to be identified in agencies' strategic plans.

The extreme left and extreme right of the model are easiest to define. The hard part is to develop measures for outputs (although those are easiest) and especially for outcomes. How would you know you are making progress toward achieving your long-term goal before you get to that goal? What would tell you that you are on the right or wrong track? How would you know whether you need to make course corrections to get to the destination you want? In developing outcome measures, keep asking like a four-year-old child, "Why? … Why? … Why?"

The further away from outputs you measure, the more likely that conditions outside the agency's/program's control are affecting the results observed. Factors such as the economy or the weather can affect long-term outcomes. And that is where third-party evaluation can be helpful to analyze the performance data, as well as other quantitative and qualitative information, to assess the impact of the agency/program on the outcomes.

The benefits of using a logic model are numerous:

- It is the strategic plan on a page. The measures can be derived directly from a logic model.
- A logic model can be a highly effective tool for communicating with stakeholders and for making sure that the activities, outputs, and outcomes are accurate in terms of their mission and business. Program people seem to "get it," and they help refine the model very quickly.
- It makes the connection between inputs, activities, outputs, and outcomes transparent and traceable.
- Most important, it shows in a nutshell where you want to go.

Related Problems and Pitfalls

Clients tend to focus on outputs, not outcomes. Output measures are much easier, and they are under the agency's control. People know what they do, and they are used to measuring it. "I produced 2,500 widgets last year" or "On the average we provided two-second turnaround time." They can find it harder to answer the question: "So what?" They are not used to looking at the outcomes, or impact, of what they do. We need to keep asking "So what?" or "Why?" Move toward what would show impact or progress toward solving the problem the agency, program, or project was created to address.

Goals and objectives are often lofty, and not really measurable. "The goal is to conduct the best census ever." How do you measure that? Make the goals concrete enough that we can know whether they have been achieved and whether we are making progress toward them.

There are tons of reports with measures that are ignored; no one knows how to use them. There is no plan to actually use the measures to make decisions about resource allocation. This is where agencies need to move from performance measurement to performance management, using the performance data to make resource allocation decisions based on credible evidence and including evaluations, analysis of agency culture, new directives from Congress or higher levels of the Administration, etc.

The client wants to use measures that they already produce, regardless of whether those are actually useful, meaningful, or important. This is the argument that "we already report performance data and have been doing it for years." These are probably outputs, not outcomes, and even so, they need to be reviewed in light of the strategic goals/objectives to determine whether they show progress toward achieving end outcomes.

They want to identify a budget as an output or outcome. A budget is always an input. Just don't let the conversation go there.

Best Practices and Lessons Learned

You need clear goals/objectives to even begin to start developing performance measures. Without clear goals, you can only measure activities and outputs. You can show, for example, how many steps travelers have taken along a path, how much food was consumed, and how long they have been traveling. But you cannot know whether they are any nearer their destination unless you know the destination. They might be walking in circles.

Performance measures are derived from strategic plans. If the agency does not have a plan, it needs to develop one. There is much guidance and many examples on developing a plan.

Complete a logic model for the whole program. You can develop outcomes or measures as you go or wait until the end, but the measures help keep the goals/objectives and outcomes real.

To the maximum extent possible, ground the logic model in bedrock. Bedrock includes the following, in the priority listed: Legislation, Congressional committee reports, executive orders, regulations, agency policies, and agency guidance. Legislation is gold. The Constitution is platinum (e.g., the requirement for a decennial census).

Long-term outcomes, or impact, are relatively straightforward to identify. It should reflect the problem that the program, agency, or project was created to solve. That is what you are trying to measure progress toward. If your program was created to address the problem of homelessness, the long-term outcome is a reduction of homelessness, regardless of how you decide to measure it.

Use caution in interpreting what the measures show. Performance measures tell you what is happening; they do not tell you why something is happening. You need to plan for periodic evaluations to get at causality. It is possible that your program kept things from being worse than they appear or that the results measured might have happened even without your program.

Fewer is better; avoid a shotgun approach to creating measures. Agencies tend to want to measure everything they do rather than focus on the most important few things. Goals might need to be prioritized to emphasize the most important things to measure.

Look at similar agencies and programs for examples of performance measures. Two types of outcomes are particularly difficult to measure: (a) prevention and (b) research and development. How do you measure what did not happen, and how do you measure what might be experimental with a limited scope? The solution for the first is to find a proxy, and the best place to look might be at similar programs in other agencies. The Department of Health and Human Services does a lot of prevention work and is a good place to look for examples. The solution to the second often takes the form of simply finding out whether anyone anywhere is using the results of the research.

The organization responsible for an agency's performance should be closely aligned with the organization responsible for its strategic planning. Otherwise, strategic plans and/or performance reports tend to be ignored. Performance management operationalizes an organization's strategic plan.

More frequent reporting tends to be better than less frequent. Agencies often have a hard time getting their performance reports done on an annual basis, and the data is so out of date that it is not helpful for resource allocation decisions. The current OMB Director is calling for performance reporting more often than weekly, which seems daunting for agencies that have trouble reporting annually, but it could actually become easier if processes are put in place to streamline reporting the few most important data items. This frequency is already being required for reporting under the Recovery Act.

Efficiency is output divided by input; effectiveness is outcome divided by input. You need a denominator in both cases to achieve these common measures of program performance. Efficiency is about producing more output with less input, but efficient does not always mean effective. Effectiveness is about results and therefore uses outcome measures. The logic model helps show clearly those relationships.

See the SEG articles "Earned Value Management" and "Acquisition Management Metrics" in the Acquisition Systems Engineering section for related information.

References and Resources

1. Advanced Performance Institute, *Strategic Performance Management in Government and Public Sector Organizations*.

2. MITRE-supported performance measurement products:
U.S. Census Bureau, August 18, 2009, Strategic Plan for the 2020 Census, Ver. 1.0.
United States Visitor and Immigrant Status Indicator Technology Program, April 16, 2009, US-VISIT Strategic Plan 2009–2013.
3. Steinhardt, B., July 24, 2008, Government Performance: Lessons Learned for the Next Administration on Using Performance Information to Improve Results, Testimony (statement) before the Subcommittee on Federal Financial Management, Government Information, Federal Services, and International Security, Committee on Homeland Security and Governmental Affairs, U.S. Senate, U.S Government Accountability Office, GAO-08-1026T.
4. The MITRE Institute, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1, p. 46.
5. U.S. Commodity Futures Trading Commission, Commodity Futures Trading Commission Strategic Plan 2007-2012, accessed February 24, 2010.
6. U.S. Government Accounting (now Accountability) Office/General Government Division, May 1997, Agencies' Strategic Plans Under GPRA: Key Questions to Facilitate Congressional Review, Ver. 1, GAO/GGD-l0.l.16.
7. U.S. Government Accountability Office, May 2003, Program Evaluation: An Evaluation Culture and Collaborative Partnerships Help Build Agency Capacity, GAO-03-454.
8. U.S. Government Accountability Office, May 2005, Performance Measurement and Evaluation: Definitions and Relationships, GAO-05-739SP.
9. W.K. Kellogg Foundation Logic Model Development Guide.

Enterprise Technology, Information, and Infrastructure

Definition: *Enterprise technology, information, and infrastructure refers to information technology (IT) resources and data shared across an enterprise—at least across a sponsor's organization, but also cross-organizational (multi-agency, Joint/DoD). It includes such efforts as infrastructure engineering for building, managing, and evolving shared IT; IT or infrastructure operations for administering and monitoring the performance of IT services provided to the enterprise; IT services management; and information services management. IT strategy and portfolio management and IT governance help the concept function effectively.*

Keywords: *information and data management, IT infrastructure, IT service management, service management*

Context

Former U.S. Chief Information Officer (CIO) Vivek Kundra's February 2011 paper on Federal Cloud Computing [1] states: "Cloud computing describes a broad movement to treat IT services as a commodity with the ability to dynamically increase or decrease capacity to match usage needs. By leveraging shared infrastructure and economies of scale, cloud computing presents federal leadership with a compelling business model. It allows users to control the computing services they access, while sharing the investment in the underlying IT resources among consumers. When the computing resources are provided by another organization over a wide-area network, cloud computing is similar to an electric power utility. The providers benefit from economies of scale, which in turn enables them to lower individual usage costs and centralize infrastructure costs. Users pay for what they consume, can increase or decrease their usage, and leverage the shared underlying resources. With a cloud computing approach, a cloud customer can spend less time managing complex IT resources and more time investing in core mission work."

Despite this endorsement, IT management and the associated shift toward common, shared resources are large concerns for many of our sponsors. MITRE system engineers (SEs) are increasingly supporting sponsors who are in the process of procuring new IT systems, migrating existing IT-based systems to a common or shared infrastructure, or upgrading their own internal business systems. Although most aspects of this shift are technical, we are recognizing that many are non-technical, and our systems engineering skills need to expand to address those aspects (i.e., governance, increased complexity of sharing resources across organizations, data ownership, service management. and life cycle).

In addition, at the center of this shift are data (or information) and the need to share it appropriately. Data is the "life blood" of an organization—as it flows among systems, databases, processes, and people, it carries with it the ability to make the organization smarter and more effective. The migration toward shared IT resources needs to accommodate the intended business operations being supported as well as the data usage, including appropriate access control and protection.

MITRE SE Roles and Expectations

MITRE systems engineers are expected to understand the systems engineering principles to be applied to the enterprise-level IT programs they support. They are also expected to understand the larger enterprise context in which the programs operate. For a particular enterprise-level program, MITRE may be asked to play a role in helping the customer define or refine business processes, such as technical or systems engineering aspects of portfolio management, or operational constructs for shared infrastructure. For mid- and senior-level MITRE staff, the role often involves recommending how to apply engineering analysis, advice, processes, and resources to achieve desired portfolio-level outcomes. Understanding the interconnections and dynamics across the different levels of an organization or multi-agency governance structure is important to providing thoughtful, balanced recommendations.

Enterprise-level efforts require many skills. MITRE SEs may be expected to support enterprise architecture, technical evolution, preliminary design of data centers or infrastructure components, implementation, monitoring and operations of infrastructure, and technical governance. Critical areas of focus normally include information assurance, data strategy, interoperability, application integration, information exchange, networks, and communications services (voice, video, and data). MITRE SEs may assist sponsors with initiatives for application migrations, infrastructure upgrades, and consolidation of computing infrastructure. Other skills involve quantifying the performance across enterprise resources and enabling service-level agreements. In cases where deep, focused technical knowledge is required, MITRE SEs must to be able to identify the need and bring in the required skills to match the challenge at hand.

Best Practices and Lessons Learned

In complex environments such as enterprise-level IT programs, three important factors should be taken into consideration: the stakeholders, the technology, and the mission the IT supports. Failure in even one of these factors can cause total program failure.

Know the stakeholders. An "enterprise" usually involves a set of constituents with various goals, requirements, and resources. Sometimes, these constituents' considerations are at odds with one another. Vital elements of the non-technical aspects of enterprise IT are understanding the various stakeholders and being able to articulate needs from their perspective. Several methods exist for analyzing stakeholders. For instance, a simple POET (Political, Operational, Economic, Technical) analysis can be used to clearly articulate issues associated with stakeholders (see the "Stakeholder Assessment and Management" article). Understanding the kind of governance required to make an enterprise function is also necessary (see the "IT Governance" article). Governance is what enables the stakeholders to communicate their needs and participate in the enterprise definition, evolution, and operation. The need for strong governance cannot be overstated.

Know the technology. A wide array of technology is associated with enterprise IT programs, from networking details to cloud computing and data centers. Keeping abreast of the current trends in the appropriate areas of your IT program allows you to address disruptive technology concerns and apply sound technical practice to the job. Because computing is so prevalent in today's society and it takes many forms from desktop PCs to handheld mobile devices, everyone touches technology and has expectations from it—often unrealistic. Our sponsors and other program stakeholders are no different. The key to managing technical expectations is knowing the technology and its applicability and having the trust of the sponsor so you can help them recognize when something is too immature for implementation and not a shrink-wrapped, off-the-shelf bargain.

In addition to knowing the technology itself is knowing how to apply good IT management techniques. IT service efforts have frameworks and best practices to leverage. A fairly complete and commonly referenced framework is the Information Technology Service Management and Information Technology Infrastructure Library (ITIL) [2]. The "IT Service Management (ITSM)" article details this further. In addition, NIST [3] provides many useful references for IT, cloud computing, security, and the Federal Information Security Management Act.

Know the mission being supported. It is very important to understand the mission(s) that the infrastructure supports. The ability to articulate the technical implications of mission needs is arguably the most valuable systems engineering talent to bring to bear on customer programs. Enterprise technology succeeds by anticipating end-user needs and proactively addressing them, not waiting for breakage or unhappy users to complain that they are not being supported. This is a complex and difficult thing to do for an enterprise, but it is necessary as computing and infrastructure become more commoditized. The Engineering Information-Intensive Enterprises section of this guide addresses ways to support the mission through enterprise systems engineering.

Articles Under This Topic

"IT Infrastructure Engineering" provides insight into the complexities of developing, managing, and operating IT infrastructure (networks and communications equipment, data centers, shared computing platforms, etc.) within an enterprise environment.

"IT Service Management (ITSM)" describes frameworks, processes, and models that address best practices in managing, supporting, and delivering IT services.

"Information and Data Management" includes best practices and lessons learned for engineering enterprise data and information.

"Radio Frequency Spectrum Management" discusses the analytical, procedural, and policy approaches to planning and managing the use of the electromagnetic spectrum.

References and Resources

1. Kundra, V., U.S. Chief Information Officer, Federal Cloud Computing Strategy, February 8, 2011.
2. IT Infrastructure Library (ITIL) OGC website, http://www.itil-officialsite.com/AboutITIL/WhatisITIL.aspx.
3. National Institute for Standards and Technology (NIST), www.nist.gov.

Definitions: *Infrastructure engineering builds, manages, and evolves the environment supporting the processes, physical and human resources needed to develop, operate, and sustain IT. Infrastructure operations address day-to-day management and maintenance of IT services, systems, and applications—plus their infrastructures and facilities. Processes include systems and network administration, data center operations, help desks, and service-level management.*

Keywords: *cloud computing, continuity of operation, data center, data center operations, disaster recovery, end-to-end computing infrastructure, IT infrastructure, servers, service management, storage area networks, unified communications, virtualization, wide area networks*

ENTERPRISE TECHNOLOGY, INFORMATION, AND INFRASTRUCTURE

IT Infrastructure Engineering

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are challenged with the rapid changes in the emerging technology of IT infrastructure. They are expected to support architecture, preliminary design, analysis, implementation, and operations of the infrastructure. Critical areas of focus include information assurance, data strategy, interoperability, application integration, information exchange, networks, and communications services (voice, video, and data). MITRE SEs assist sponsors with initiatives for data centers, application migrations, infrastructure architecture, and consolidation of computing infrastructure. MITRE SEs develop competencies in data center operations, infrastructure platforms, and IT service delivery. Technical specialties to which they should reach back include local and wide-area network design, servers, storage, backup, disaster recovery, continuity of operation (COOP), performance monitoring, virtualization, cloud computing, modeling, visualization, voice over Internet protocol (VoIP), IPv6, and other emerging technologies.

Background

MITRE SEs are expected to take a total life-cycle approach to assist operational users in applying IT infrastructure, operations, maintenance, and management techniques to meet their challenges.

Infrastructure Engineering and the associated Operations and Service Management expertise includes:

- Implementation of Information Technology Service Management and Information Technology Infrastructure Library (ITIL) concepts and policies (for more details, see the article, "IT Service Management (ITSM)" under this topic)
- Development of infrastructure strategy and IT operational policies, standards, and processes tailored to agency o-r department missions
- Development of infrastructure and operational requirements in all phases of the system development life cycle
- Development of asset management processes that support the provisioning, tracking, reporting, ownership, and financial status of IT assets
- Data center operations, consolidations, and relocations; planning, implementing, and testing for disaster recovery; daily operations and data center management, including server and systems migrations
- Service desk, help desk, and contact and call center development, implementation, operations, and process improvement
- Service-level management through the development of processes, people, technology, and service-level and operating-level agreements
- Technical strategy, architecture, and design incorporating emerging technologies such as virtualization, cloud and utility computing, IP telephony, and IPv6 planning and migration
- Infrastructure and operations security, such as network and application firewalls, authentication, identity and privilege management, and intrusion detection and prevention
- Beyond technical deliverables, assist with various road shows, technical exchange meetings, and conferences to promote the importance of a solid infrastructure

Government, Industry, and Commercial Interest in IT Infrastructure

In December 2010, the U.S. Federal Government Chief Information Officer released a 25 Point IT Management Reform Plan that concentrates on areas to reduce IT operating costs and to bring greater value through IT consolidation. The emphasis is on reducing data centers and migrating to lean and agile IT computing services [1].

The National Institute of Standards and Technology (NIST) took the lead to define cloud computing in the context of cost savings and "increased IT agility." This effort provided the momentum to challenge the rising and unsustainable costs in response to "difficult economic constraints." NIST is partnering with all stakeholders (including MITRE) to face the challenges of security, privacy, and other barriers that have hindered a broad adoption of cloud-based IT infrastructure [2, 3].

The U.S. General Services Administration (GSA) sought and adopted lightweight and agile IT infrastructure to support their common enterprise infrastructure (e.g., enterprise email) while reducing the costs and increasing efficiency of the associated acquisition and deployment. Additionally, GSA is taking a lead role in deploying Software as a Service (SaaS) through the apps.gov portal [4]. This effort emphasizes compliance with Certification and Accreditation and FISMA [5] Moderate Impact Data security requirements prior to loading their applications to the store for distribution.

Best Practices and Lessons Learned

Translating business objectives into IT infrastructure needs. The most difficult part of infrastructure engineering is identifying the infrastructure requirements implied by the sponsor's business objectives. Business objectives, by definition, are not technological. Deriving the technical requirements for the IT infrastructure needed to support business objectives is a critical technical contribution. For example, translating a business need for enhanced distributed capabilities may require the development of a Network Design guide where the technical principles for switching (e.g., VLANs, Ethernet, STP), routing (e.g., RIP, EIGRP, OSPF, ISIS, BGP), Quality of Service (QOS), and wiring/physical infrastructure are mapped to the business objectives. By creating such a guideline, the client is then able to make technically supported decisions to meet their objectives.

Governance. Because infrastructure supports the entire range of an enterprise's IT needs, it requires a broad level of coordination. Every department and function in the enterprise needs to be represented in the governance of the infrastructure. Plan for significant investment of time and resources in governance boards, outreach programs. and socialization of change. (For more details on governance, see the SEG articles, "Enterprise Governance," "IT Governance," and "Transformation Planning and Organizational Change.")

Infrastructure evolution. Infrastructure Engineering is distinguished from other IT efforts by the almost absolute necessity of incremental evolution. It is extremely rare for an enterprise to be able to switch from one infrastructure to another in one fell swoop. Plan and organize based on incremental change. Provision for operating both old and new infrastructure components in parallel. (For more details, see the articles on Configuration Management.)

Service-level agreements. Because the infrastructure supports the entire enterprise, it is impractical and inappropriate to organize interfaces around traditional interface control documents. Users (and potential users) of an infrastructure or shared core function demand a different kind of performance guarantee based on the one-to-many relationship between the owners of the infrastructure or shared function and their customers. This guarantee is captured in a service-level agreement (SLA) that documents the expected performance and behavior of the infrastructure for use. Because the SLA is, in effect, an internal contract between the infrastructure and its users, Infrastructure Engineering must provide for precise measuring, monitoring, and reporting of the function's behavior in the design and in the operation—to the degree that the SLA can be enforced. This requires significantly more detail and rigor than is usually applied to just developing an infrastructure by itself.

Versioning and provisioning. Our sponsor's enterprise is usually large, complex, and widely distributed. As a consequence, it is virtually impossible to change every physical instance of an infrastructure component at one time. Plan for operating multiple versions of any infrastructure component being updated or replaced. It is common for a physically distributed enterprise to be operating two, three, or even four different versions of a single component at the same time. Account for multiple versions, not just for brief periods but continuously as the infrastructure evolves. (For more details, see the articles under Configuration Management in the Acquisition Systems Engineering section.)

Baseline infrastructure assessment. Assessing an operational environment is often a first step in an infrastructure engineering effort. The focus of the assessment should be based on the customer needs and requirements. Two examples are:

- Assess a baseline configuration of an existing operational environment to use for gap analysis of an "AS-IS" versus a "TO-BE" architecture.
- Compare a baseline configuration of an existing operational environment against a secure configuration standard for a security assessment.

Common security processes. Perform trusted, independent vulnerability assessments to highlight issues and help remedy and mitigate risk based on NIST, NSA, and leading industry practices in the information assurance and security realm. Document security vulnerabilities and provide recommendations for resolution, mapping the findings to NIST 800-53 [6] controls and providing a risk level report. Promote a standard set of commercial tools such as NetDoctor, Nessus®, or Wireshark where applicable. These tools reuse a "Findings Dictionary" to document common vulnerabilities and provide a consistent approach across assessors and assessment organizations—multiple systems engineers from different organizations can all perform the same science, technology, and engineering for different customers in the enterprise following the same documented processes.

Technology transition testing. Leverage the effort of industry experts by partnering with accredited test laboratories. For example, preparing for changes to computer networks to support the IPv6 addressing plan requires a partnership with NIST, federal agencies, or government entities, and the wide range of commercial network equipment vendors. The IPv6 Transition effort is based on a "target architecture" to focus on operational testing. Test planning includes implementing a test laboratory architecture, proving out operational Dual Stack configurations, and identifying testing requirements for pilot deployment.

Next-generation network—the evolution continues. Network technologies and capabilities continue to evolve with the continued growth of the Internet. The current trend toward converged services is apparent and seen across the federal government. This shift requires a robust core and reliable end-to-end services at a minimum. Key next-generation network infrastructure attributes include:

- Robust core technologies:
  - Multiprotocol label switching
  - High-end routers/switches
- Convergence:
  - Voice, video, data on a single infrastructure
  - Broadband wireless access (4G/3G)
  - Mobile applications and value-add services and applications are drivers
  - Carrier class devices
  - Network is transparent to end user
- Multi-platform, multimedia, multi-channel, multi-purpose platforms—Android, Blackberry, iPhone, iPad, and Windows platforms
- Security-centric: Sensitive and critical information riding on a single infrastructure requires SLA and carrier class devices/services.
- Low cost: Economies of scale are pushing a low-cost model approach:
  - Virtualization and cloud
  - Infrastructure consolidation
  - Green IT
- Unified communications: More than just VoIP:
  - Video teleconference, teleconference, virtual meeting spaces
  - E-boarding and collaboration
  - Presence and mobility
  - Platform and technology agnostic
  - IP telephony

An efficient infrastructure. Assess cabling, power, grounding, heating, ventilation and air conditioning, raised flooring, load bearing, fire suppression, physical access and egress (ADA compliance). They follow applicable local codes and ordinances, using the ANSI-EAI, NEMA, and NEC as references, and create recommendations for sponsors to follow based on standards. Currently, "green" initiatives cost more than standard infrastructure build-outs; however, when life-cycle costs can be shown to be equal (or less) based on operating savings (i.e., lower electric bill due to increased efficiencies), the effort to move to a green infrastructure may be justified. (For more details, see the articles under Integrated Logistics Support in the Acquisition Systems Engineering section.)

Mobile IT management and support. Mobile IT Platform diversity complicates IT management and help desk activities because these platforms are incompatible. IT departments need to revise processes for developing applications to accommodate the new workflow and mobile data platforms. Evolving security policies and blurred lines between the personal and professional role of wireless devices require security approaches that go beyond traditional firewalls. Most enterprise infrastructure architecture mapping efforts focus on fixed IT assets and core applications that run on them. Mobile devices and applications are often unaccounted for in future plans of architectures. Required infrastructure engineering capabilities include:

- Mobile Technology Policy/Security Development Support
- Mobile IT System Design Support
- Mobile IT System Integration Support
- Mobile IT Change Management Support
- Mobile Workforce Management Support
- Mobile IT Performance Management Support

References and Resources

1. 25 Point Implementation Plan to Reform Federal Information Technology Management.
2. The NIST Definition of Cloud Computing, NIST Cloud Computing Program, NIST.
3. Cloud-Based Infrastructure as a Service Comes to Government, GSA.
4. U.S. GSA Apps.Gov Portal, GSA.
5. Federal Information Security Management Act (FISMA) Implementation Project, NIST.
6. Recommended Security Controls for Federal Information Systems and Organizations, NIST.

Additional References and Resources

Hoskins, J., 2004, *Building an On Demand Computing Environment with IBM: How to Optimize your Current Infrastructure for Today and Tomorrow,* Maximum Press.

Foster, I., and Kesselman, C., eds., 2004, *The Grid 2: Blueprint for a New Computing Infrastructure*, Elsevier.

Sasaki, R., 2005, *Security and Privacy in the Age of Ubiquitous Computing*, International Federation for Information Processing.

Definition: *Information Technology (IT) Service Management is a generic umbrella for frameworks, processes, and models that address best practices in managing, supporting, and delivering IT services. IT services may include (as defined by NIST for cloud computing): Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).*

Keywords: *CMM, COBIT, infrastructure services, ISO 20000, ITIL, ITSM, service delivery, service desk, service management, service support*

ENTERPRISE TECHNOLOGY, INFORMATION, AND INFRASTRUCTURE

IT Service Management (ITSM)

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) supporting sponsors procuring new IT systems, migrating existing IT-based systems to common or shared infrastructure, or upgrading their internal business systems must have an understanding of the IT and associated processes for control, operations, shared use, and governance. MITRE SEs develop comprehensive programs around an ITSM framework or approach or address specific issues in particular process areas. Examples include developing implementation plans for migrating from decentralized help desks to centralized service desks, recommending process improvements to improve overall system availability, designing end-to-end monitoring systems, developing service-level agreements, and identifying critical support factors for service management process areas.

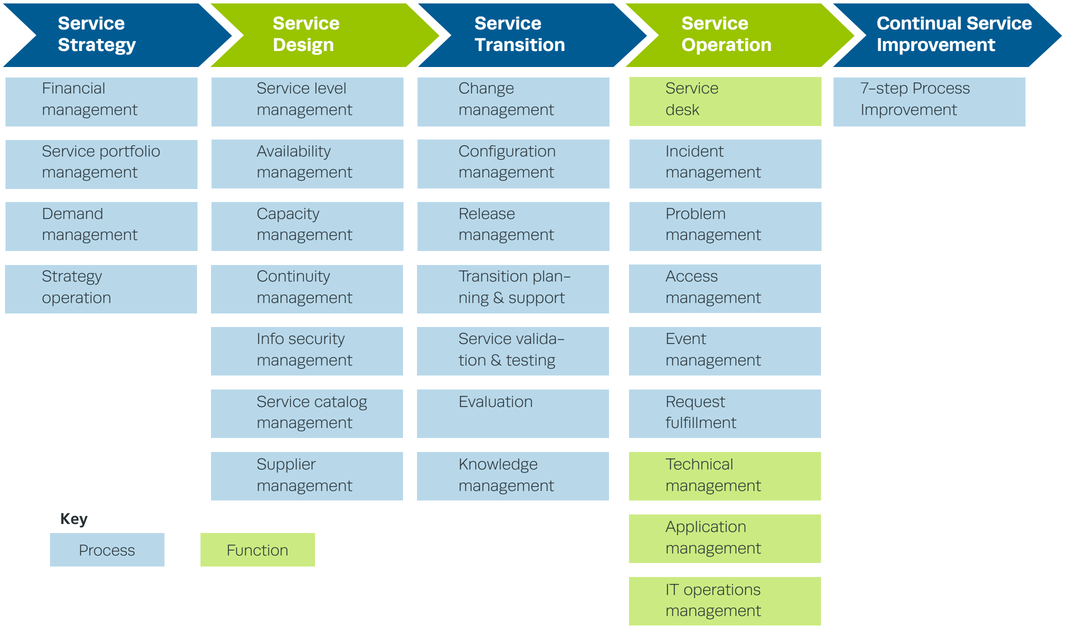
Background

IT service providers have several frameworks and best practices to leverage, whether they are commercial providers or internal IT organizations. This article focuses on the processes and practices defined in the Information Technology Infrastructure Library (ITIL), by far the most comprehensive and widely adopted framework for IT service management.

Other related best practice frameworks for ITSM include:

- **Control Objectives for Information and related Technology (COBIT),** which was developed by the ISACA IT Governance Institute.
- **IT Service Capability Maturity Model (CMM, CMMI, CMM Services)**, which provides standards for defining process areas and measuring an organization's level of maturity within these process areas.
- **ISO/IEC20000**, which is an international standard, based on the ITIL Framework, for IT organizations that want formal certification of their service management processes.

The ITIL is a framework developed by the United Kingdom Office of Government Commerce (OGC). The original ITIL framework was developed in the late 1980s and was then documented in a collection of books in the 1990s known as ITIL v2. The current version of the framework, ITIL v3 was released in 2006.



Service Strategy	Service Design	Service Transition	Service Operation	Continual Service Improvement
Financial management	Service level management	Change management	Service desk	7-step Process Improvement
Service portfolio management	Availability management	Configuration management	Incident management	
Demand management	Capacity management	Release management	Problem management	
Strategy operation	Continuity management	Transition planning & support	Access management	
	Info security management	Service validation & testing	Event management	
	Service catalog management	Evaluation	Request fulfillment	
	Supplier management	Knowledge management	Technical management	
			Application management	
			IT operations management	

Figure 1. ITIL v3 Phases and Process Areas

The ITIL framework is based on the concept that IT organizations provide services, not technologies, to business customers. The difference between a service and a technology is the added value IT organizations provide to their customers by accepting and managing the risk associated with that technology. In simple terms, an exchange server is a technology. Electronic mail or messaging is a service that includes support and management functions whose details are hidden from the end user or customer. ITIL divides these support and management functions into 30 process areas that arise through the service life cycle.

Although the ITIL does describe the major elements of the 30 process areas that are considered to be best practice, the ITIL is not prescriptive. For example, the ITIL describes Continual Service Improvement processes but does not require the use of one process improvement methodology over another. Practitioners can rely on Lean, Six Sigma, or other methodologies for process improvement. Similarly, the ITIL allows for the use of COBIT, for example, for the change management and governance related processes, and has a similar, complementary relationship with CMM. The ITIL also provides descriptions of roles associated with each of the process areas that should be considered within the governance structure.

The service life cycle is divided into five phases, each with its set of process areas that play critical roles during that phase (see Figure 1). Note that some of the process areas mentioned in the ITIL body of knowledge are relevant throughout the service life cycle. The classification of processes under a specific phase is only meant to demonstrate when these process areas are most important.

Why Implement ITIL?

The ITIL provides a framework for viewing IT support and service processes. None of the ITIL process areas is “new” or different from the traditional IT process areas. What is different about the ITIL is the acknowledgment that IT is no longer driving business decisions. To the contrary, IT services have largely become a commodity. This shift has often caused IT organizations to become separated and marginalized from the business operations they support. The ITIL framework was designed with the objective of injecting IT back into business decisions; that is, the ITIL aims to reestablish the participation of IT in making business or mission decisions with the goal of delivering relevant, improved services at a reasonable cost. By involving IT at the beginning of the service life cycle, support and service level offerings can become a standard part of every IT service.

Poor-performing IT operations is often a symptom of the problem, but not the problem itself. IT operations receive systems from business units, applications development, systems engineering, and other parts of the organization. Lack of organizational processes and standards can cause IT operations groups to have to manage every version of every platform and application available on the market. Often this is a consequence of an organizational business model in which IT operations has no voice in the decision making for the design of systems that they later own after the transition portion of the life cycle. The earlier in the design phase life-cycle management (or sustainment) is built in, the more likely the overall cost and performance objectives can be achieved. This is a critical and often overlooked point.

The ITIL helps point to the processes that begin from the conceptualization phase of a new system, continue through acquisition, and then move to change, configuration, and release management processes that directly impact application development and systems engineering teams. Most important, the processes include mission/business representatives as an integral part of the service development process.

Finally, the ITIL stresses the importance of metrics, both in measuring the success of the ITIL program itself and for measuring the performance of the IT organization in delivering customer services. Because ITIL programs are often lengthy, it is critical to demonstrate improvements throughout the duration of the program.

Best Practices and Lessons Learned

During the early 2000s, ITIL became a popular framework for IT organizations to adopt, including those within federal government agencies. However, federal government agencies are still catching up with the private sector in implementing ITIL.

Are we there yet? Implementing an IT Services Management framework is a lengthy process. Organizations can expect to spend up to two years on these efforts, even if they focus on just a subset of the ITIL process areas. For this reason, ITIL programs require senior leadership buy-in in order to be successful. Strong governance is a key component of even limited success.

It's not just about the IT. IT services management extends beyond IT operations and into all aspects of IT services, including acquisition planning, financial planning, portfolio management, and release management. Don't make the mistake of focusing IT services efforts only on IT operations. As noted, operational performance issues are usually the symptom, not the root cause of the problem.

Are you being served? Often IT organizations are hesitant to include representatives from outside of their organization in their IT services efforts. Instead, they focus exclusively on internal IT process improvement efforts. This misses the whole point of IT services management, which is to view stakeholders and especially customers or users as partners. The shift toward IT's being a commodity means that bringing the customers or users into the project translates to better understanding their needs and level of service required.

Measuring business value. Defining metrics for an IT services program is often overlooked. It is not always obvious that an improvement in change management can directly impact availability of critical systems. Metrics need to be closely tied to the strategic goals and value of the IT program, and they need to be relevant to the business or mission being supported. Metrics need to be defined, collected, and shared throughout the program. Good sources of material on metrics useful for IT can be found on Gartner, Corporate Executive Board, and CIO Executive Council websites (access is for members).

References and Resources

Jan van Bon, Tieneke Verheijen, 2006, *Frameworks for IT Management*, Van Haren Publishing

Office of Government Commerce, 2007, ITIL v3 Library, TSO. (Book)

ISO/IEC 20000-1:2011, Information technology—Service management—Part 1: Service management system requirements

NIST Special Publication 800-145 (Draft), The NIST Definition of Cloud Computing.

Definition: *Information and data management (IDM) forms policies, procedures, and best practices to ensure data is understandable, trusted, visible, accessible, optimized, and interoperable. IDM includes processes for strategy, planning, modeling, security, access control, visualization, data analytics, and quality. Outcomes encompass improving data quality and assurance, enabling information sharing, and fostering data reuse by minimizing data redundancy.*

Keywords: *business intelligence, data, data analysis, data governance, data management, data mart, data mining, data modeling, data quality, data warehouse, database, database management system (DBMS), information management, master data management, metadata, data migration*

ENTERPRISE TECHNOLOGY, INFORMATION, AND INFRASTRUCTURE

Information and Data Management

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) will encounter IDM-related activities on most government programs. They are expected to understand the customer organization's data requirements and help develop concepts for how to use and manage data, as well as how to apply appropriate IDM mechanisms in the organization's system environment. The IDM SE role may start before system acquisition, when only general requirements are known. Typically it encompasses planning, training, and operational support for the awareness, coordination, and integration of data and information management activities. MITRE SEs are expected to be able to determine the size of data, data security and privacy requirements, and data sharing requirements. This may include specifying the information needs, data, software, and hardware, as well as the skills and staffing required to support the system's operational IDM needs. At the end of a system life cycle, the SE may need to consider where and how data is stored or disposed.

Discussion

Data is the "life blood" of an organization, for as it flows between systems, databases, processes, and departments, it carries with it the ability to make the organization smarter and more effective. The highest performing organizations pay close attention to the data asset, not as an afterthought but rather as a core part of defining, designing, and constructing their systems and databases. Data is essential to making well-informed decisions that guide and measure achievement of organizational strategy. For example, an organization may analyze data to determine the optimal enforcement actions that reduce non-compliant behavior. Similarly, data is also at the heart of the business processes. An organization may enhance a process to catch fraudulent activities by including historical risk-related data. Over time, this type of process improvement can result in material savings. Even a single execution of a business process can translate into substantial benefits, such as using data patterns to stop a terrorist at a border or filtering a cyber attack.

How an organization uses and manages the data is just as important as the mechanisms used to bring it into the environment. Having the right data of appropriate quality enables the organization to perform processes well and to determine which processes have the greatest impact. These fundamental objectives leverage data by transforming it into useful information. The highest performing organizations ensure that their data assets are accessible to the processes and individuals who need them, are of sufficient quality and timeliness, and are protected against misuse and abuse. Successfully leveraging data and information assets does not happen by itself; it requires proactive data management by applying specific disciplines, policies, and competencies throughout the life of the data.

Similar to systems, data goes through a life cycle. Figure 1 presents the key phases of the data life cycle.

Effective data management through all of the data life-cycle phases is the foundation for reliable information. Data may have

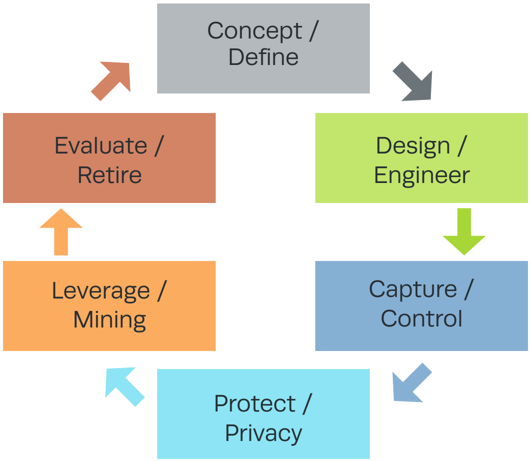


Figure 1. Data Life Cycle

different uses at different times and requires different management handling in the life-cycle phases. For instance, an organization may consider critical data required for discovery as very valuable during a key event, but when the event is over, the information diminishes in value quickly (e.g., data collected for predicting the weather).

Data may typically have a longer lifespan than the project that creates it. Though the funding period formally defines the lifespan of most projects, the resultant data may be available for many years afterwards. If an organization manages and preserves the data properly, the data is available for use well into the future, increasing the investment made in generating it by increasing visibility and usefulness. The time spent in planning and implementing effective data management pays dividends far in excess of its investment costs.

IDM is the set of related disciplines that aims to manage the data asset fully, from conception to retirement. Figure 2 presents a high-level view of data management disciplines.

Data without context has no value; data that consumers never use is worthless, too. The value of data is in the information it contains and uses. The extraction of information and

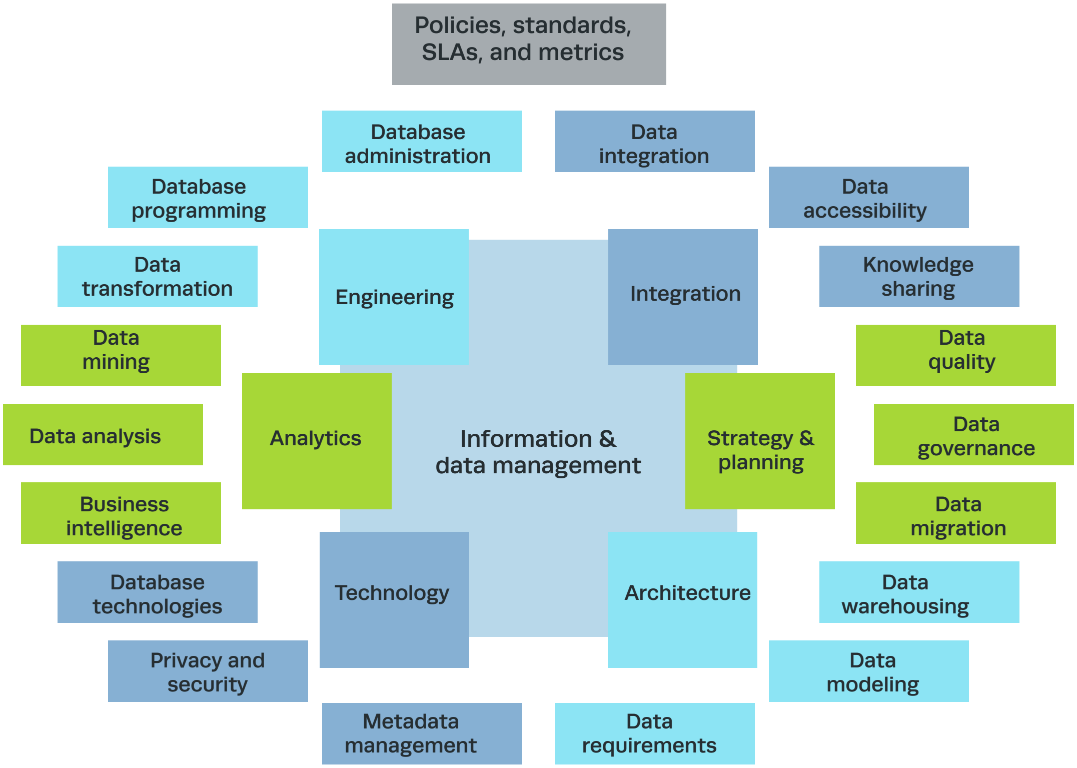


Figure 2. Data Management Disciplines

providing it in an appropriate format may be summarized as data analysis and reporting. However, data analysis and reporting encompasses several overlapping disciplines, among them statistical analysis, data mining, predictive analysis, artificial intelligence, and business intelligence. IDM has an appreciation for these disciplines and may use the same tools and incorporate some of these disciplines. The common ground among all of these disciplines and IDM is making good use of data.

Knowledge Required

A MITRE SE dealing with data should be knowledgeable in at least one of the following environments or disciplines:

- **Operational data:** Operational environments provide core transactional capabilities (i.e., processing applications, claims, payments, etc.) that typically work with database management systems.
- **Data exchange:** Organizations use data exchanges and data exchange standards to share information with internal or external parties. Standardizing exchange formats and metadata minimizes impacts to both the sending and receiving systems and reduces cost and delivery time. A related discipline is master data management (MDM). An example is a vendor list. The U.S. Treasury requires specific information identifying contractors before the federal government reimburses them. Most federal agencies use this centrally collected list. Exchange, transform, and load (ETL) tools typically support these types of data exchange activities. ETL tools manipulate data and move it from one database environment to another.
- **Data warehouses [1]:** The integration of similar and disparate data from across organizational, functional, and system boundaries can create new data assets. The organizations can use the new data to ensure consistent analysis and reporting and to enhance the information needed for decision making. Data may be structured, unstructured, or both. Business intelligence (BI) has become a recognized discipline. It takes advantage of data warehouses (or similar large data consolidation) to generate business performance management and reporting.
- **Data mining and knowledge discovery:** Mining applications explore the patterns within data to discover new insight and predictive models. An organization may use specialized software that applies advanced statistics, neural net processing, graphical visualization, and other advanced analytical techniques against targeted extracts of data. In addition, tools may evaluate continuously streaming data within operational sources.
- **Database administration [2]:** Knowledge in this discipline requires specific training related to a specific DBMS and being certified. A certified database administrator (DBA)

is responsible for the installation, configuration, and maintenance of a DBMS (e.g., storage requirements, backup and recovery), as well as database design, implementation, monitoring, integrity, performance, and security of the data in the DBMS.

- **Data architecture:** A data architect is responsible for the overall data requirements of an organization, its data architecture and data models, and the design of the databases and data integration solutions that support the organization. The data structure must meet business requirements and regulations. Good communication and knowledge of the business must be part of the data architect's arsenal. A specialized area in data architecture is the role of the data steward. The data steward is usually responsible for a specific area of data such as one or more master data.

Note that:

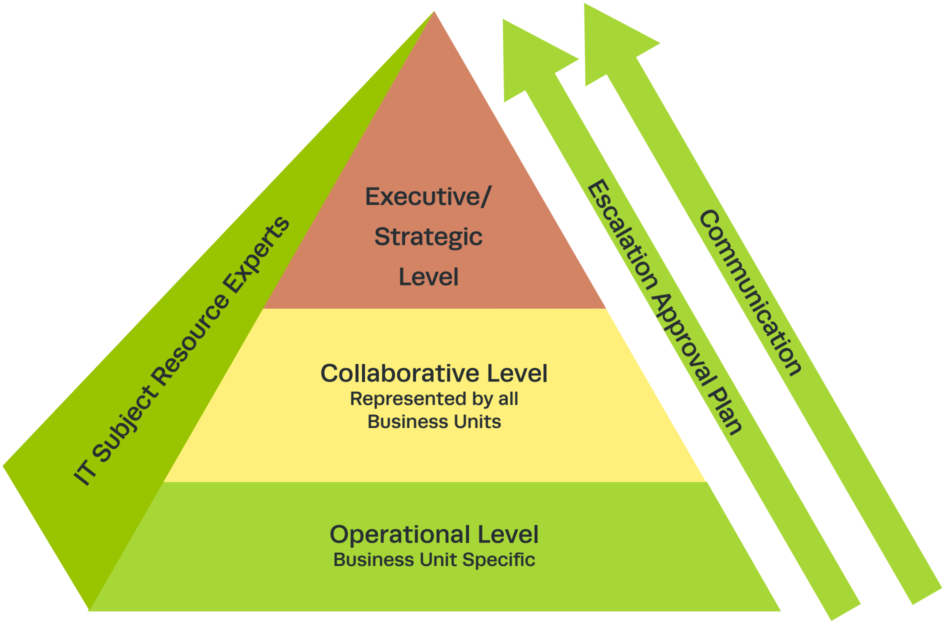
- A database is a collection of related data. It may be stored in a single or several files. It may be structured or unstructured.
- A DBMS is software that controls the organization, creation, maintenance, retrieval, storage, and security of data in a database. Applications make requests to the DBMS, but they do not manipulate the data directly.

Best Practices and Lessons Learned

What is in it for me? Conveying the importance of information and data management to federal executives is the most common challenge that an SE will encounter. Most organizations focus their time and energy on application development and the technical infrastructure. For information systems, at best this approach leads to delays in implementation and, at worst, data is not trusted and system failures occur. The organization needs to coordinate data and IT staff with the business staff to align strategy and improvement initiatives. The best approach for long-term success is to initiate a program that gradually addresses the multifaceted challenges of data management.

An effective data management program begins with identifying core principles and collaborative activities that form the foundation for providing efficient, effective, and sustainable data. The organization should interweave the following core principles throughout all of the data management activities:

- Data collected is timely, accurate, relevant, and cost-effective.
- Data efforts are cost-efficient and purposeful, and they minimize redundancy and respondent burden.
- Data is used to inform, monitor, and continuously improve policies and programs.
- Data activities seek the highest quality of data and data collection methodologies and use.
- Data activities are coordinated within the organization, maximizing the standardization of data and sharing across programs.



Executive Level

- Data Steering Committee
- Senior executives
- Chief Information Officer (CIO)
- Strategic Level
- Program Management Office

Collaborative Level

- Data Governance Council
- Data steward chair for each dataset
- IT support

Operational Level

- Operational data stewards, users
- Data steward facilitators, data definers, producers, users, SMEs, and other administra-tive support

IT Subject Resource Experts

- System/data resource experts
- IT staff, including application development, data design, security, and other data
- Resource management

Figure 3. Data Governance Framework

- Partnerships and collaboration with all stakeholders are cultivated to support common goals and objectives around data activities.
- Activities related to the collection and use of data are consistent with applicable confidentiality, privacy, and other laws, regulations, and relevant authorities.
- Data activities adhere to appropriate guidance issued by the organization, its advisory bodies, and other relevant authorities.
- The data management program supports the framework that facilitates relationships among the organization's staff, stakeholders, communities of interest (COIs), and users. It also provides a plan and approach to accomplish the next level of work needed to implement the technical architecture. The ultimate goal of the program is to define a data-sharing environment to provide a single, accurate, and consistent source of data for the organization.

Design for use. A simple analogy is to view data as a set of books. With a small number of books and only one individual who interacts with them, organizing the books is a matter of preference. Further, finding sections of interest in the books is manageable. However, as the number of books increases and the number of individuals interacting with them also increases, additional resources are required to acquire, organize, and make the books available when requested.

In the discipline of data management, acquiring, managing, and extracting information are also true for data, but at a more intricate level. The complexity of the tasks related to database 1 design grows as requirements, number of users, and data relationships increase. The most common approach to deal with large amounts of data with multiple users is to store data in a Database Management System 2 (DBMS). In many cases, the DBMS is a relational DBMS (RDBMS), which reduces reliance on software developers and provides an environment to establish data standards. However, working with any DBMS requires knowledge of the specific DBMS. In addition, an SE would have to be proficient in specific tools such as data modeling, query language, or others. A DBMS designer also would take into consideration:

- Business requirements
- Operational requirements (is it mainly an interactive system for data collection or is it for querying?)
- Access and usage requirements
- Performance
- Data structure and replications requirements
- Interfaces and data-sharing requirements
- Reporting and analytical requirements
- Data volume
- Privacy and security

The complexity of data may require both a data architect and a certified DBA. A MITRE SE may play these roles or advise someone playing these roles. A data architect is usually associated with data strategy and data modeling. The data architect may propose a physical data model, but it is in coordination with the DBA. Though the DBA's responsibilities usually start with the physical database model, their responsibilities span into all physical data responsibilities while data is in the DBMS.

Fit for consumption. The Federal Data Architecture Subcommittee (DAS) Data Quality Framework [3] defines data quality as "the state of excellence that exists when data is relevant to its intended uses, and is of sufficient detail and quantity, with a high degree of accuracy and completeness, consistent with other sources, and presented in appropriate ways." A simpler definition is "data fit for its intended use." A set of characteristics provides the best definition for data quality. These are data accessibility, data completeness, data consistency, data definition, data accuracy, data relevancy, data timeliness, and data validity. Emphasis on one characteristic over another depends on the environment. The following environments introduce key considerations:

- Stand-alone: Usually data from a single application with limited or no interfaces
- Enterprise-wide: Data of relevance to the enterprise with no interfaces to the external world
- Multi-enterprise: Data shared outside the enterprise with the need to meet external regulations

In a stand-alone environment, obtaining an acceptable level of data quality is relatively simple. The organization can meet most of the characteristics because they are part of the application requirements and design. In such a case, data quality usually means data accuracy and data validity. The organization manages the data quality by ensuring that data collection meets requirements and there are tools (automated or otherwise) to control and monitor data validity and accuracy.

The picture changes in an enterprise environment because there are competing needs for the same sets of data. For example, an accounting department must account for every penny to avoid legal consequences, whereas budgeting operations are typically not concerned with small dollar variations. In this environment, all the data quality characteristics are important, but usage determines what is acceptable and what is not. Another common factor is the variation in terminology, such as using the same word to mean two different things or using different coding lists for equivalent attributes. A recommended solution to eliminate or reduce miscommunications is to establish data stewardships and data governance to facilitate mediation and conflict management. In addition, as in most large endeavors, documentation and standards are critical for success.

The multi-enterprise environment adds complexity (i.e., data sharing). An organization may use the data in the manner originally intended. Documentation of data content is important, and control of data use is more limited, so standards are harder to enforce. As an example, the unique identification of an individual varies from state to state. A federal agency integrating data from states that do not share unique identifiers may introduce data incompatibility issues (e.g., fraud may go on unnoticed). This issue is not easily resolved because one state may mandate the use of social security number as an identifier, whereas another state may forbid it. In such a case, compromised data quality will occur until the organization implements an innovative solution that ensures uniqueness.

'Cause they said so. Data governance encompasses roles, responsibilities, accountability, policy enforcement, processes, and procedures that ensure data value, quality improvement, and standard definitions. It also entails the overall management of the availability, usability, integrity, and security of the data employed in the enterprise. A sound governance program includes a governing council, an accountability structure, a defined set of procedures, and a plan to execute those procedures. The Data Governance Framework presented in Figure 3 provides an overview of the expected governance roles and responsibilities, accountability, and authority for the strategic, collaborative, and operational levels and the IT subject matter experts.

The line of business (LOB) chief has a clear responsibility over the business. In addition, the staff at the operational level (i.e., data stewards, SMEs, etc.) receive direction from the LOB chief. Operational data stewards are responsible for managing data in the best interest of the LOB. However, when several LOBs are dealing with the same set of data, conflicts may arise because of their varying needs. Resolution of these issues requires collaboration among the LOBs. The most important role of the data governance council (or equivalent) is conflict resolution. Business and technical staffs, specifically the collaborative data stewards, should define the composition of the data governance council. The collaborative data stewards should be knowledgeable in more than one LOB as part of proposing solutions that are best for the enterprise. By promoting accountability for data as an enterprise asset and providing for efficient collaboration among stakeholders, the data governance council fosters an environment that ensures optimal mission performance. Even with the best of intentions, the data governance council may deadlock. In such cases, the collaborative steward must escalate the issues to the executive/strategic level.

Establishing a data governance council may be easy, but an effective council must be committed to collaboration. The role and responsibilities should be clear and focused to accomplish what is best for the enterprise. In some organizations, the council is composed of individuals from the LOBs, whereas in others, a separate independent group is established. Success with either approach depends on the organization.

Secure your belongings. Data security [4] protects data from unauthorized access, use, disclosure, and destruction, as well as preventing unwanted changes that can affect the integrity of data. Ensuring data security requires paying attention to physical security, network security, and security of computer systems and files. Data security is required to protect intellectual property rights, commercial interests, or to keep sensitive information safe. Security defines the methods of protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability, whether in storage or in transit. Confidentiality will prevent the disclosure of information to unauthorized individuals or systems. Integrity means that the data cannot be modified without authorization (i.e., integrity is violated when an individual accidentally or with malicious intent deletes important data files). Availability means that the information must be obtainable when a user requests the data. These three concepts are core principles of information security.

Data about data. Informative and relevant metadata (i.e., data about data) supports your organization and helps everyone that uses your data. A data steward, working under the direction of a data architect and a DBA, is usually responsible for managing a portion of the metadata. Metadata describes the definition, structure, and administration of information with all contents in context to ease the use of the captured and archived data for further use. The traditional data administration approach uses metadata to define data structures and relationships (e.g., data models) to support the development of databases and software applications. In addition to supporting systems development, metadata may be associated with all data in the enterprise for the purposes of "advertising" data assets for discovery. Organizations have to identify and document all data to facilitate its subsequent identification, proper management, and effective use, and to avoid collecting or purchasing the same data multiple times. There are many types of metadata, including vocabularies, taxonomic structures used for organizing data assets, interface specifications, and mapping tables.

Metadata management not only encapsulates basic data dictionary content but also ensures data's ongoing integrity. Metadata aids in the comprehension of the data to avoid making incorrect decisions based on their interpretations. Data lineage, the understanding of data from its inception to its current state, is a foundation capability of metadata management. As users reuse data from an original source system to the downstream support systems, they need to understand the lineage of that data. Data longevity is roughly proportional to the comprehensiveness of the metadata. For example, during an emergency event, it can be difficult to know where data is in order to assemble it expeditiously. Access to the data is critical when saving time means saving lives. Good metadata can help overcome the obstacles and get the right information into the hands of the right people as fast as possible.

Going to a better place. Data migration is the process of transferring data from one system to another. Migration includes the following steps:

- Identify the migrating legacy data and associated business rules.
- Map and match the legacy data to the target system.
- Aggregate, cleanse, and convert legacy data, as needed, to fit appropriately in the target system.
- Migrate the data in an appropriate sequence to the target system.

The most frequent challenges a data migration effort may face are an underestimation of the task and a postponement until the target system is almost ready to go operational. The complexity of a migration effort is in the implementation, and challenges exist at every step of the process. It is easy to reach Step 4 and discover that Step 1 is not complete. In some instances, legacy data cannot be migrated because it does not meet business rules in the target system and there may be a cascading effect on the cleansed data. Data cleansing is the process of detecting and correcting or removing corrupt or inaccurate records from a record set, table, or database.

Defining data elements and associated business rules can be a daunting exercise but a necessary task to ensure a successful migration effort. Legacy systems may not always document the data well, and business rules may not be fully enforced. For example, the definition of an existing data element could change midstream and affect associated business rules. Mapping may be possible, but the business rules may differ significantly to render legacy data useless. A detailed data cleansing routine will ease the pain during the tedious process of weeding out duplicates and obsolete data, as well as correcting any errors in the data.

Finally—and this is a common mistake—never assume that the previous steps worked perfectly. Routines to cleanse, transform, and migrate the data have to be run several times and at times modified to ensure completeness. The best advice for data migration is to start early in the system migration process. Be prepared. Understand as much as possible what data is available (i.e., legacy system) and where data is moving (i.e., target system). Be patient, be flexible, and expect the unexpected.

Play nice in the sandbox. Information sharing is the exchange among individuals, organizations, systems, and databases across domains and organizational boundaries. The goal of information sharing is to provide the right data at the right place in order to support timely and effective decision making. Information-sharing solutions support the collection of data from enterprise systems and their assembly into concise, understandable, actionable, and when possible, unclassified formats. An organization can have an information-sharing culture that embraces the exchange of information and an information-sharing environment that includes policies, governance, procedures, and technologies that link resources (people, process, and technology) of stakeholders to facilitate information sharing, access, and collaboration. A mature organization will exhibit continual information sharing in a standardized manner with guaranteed data quality.

References and Resources

1. The Data Warehousing Institute (TDWI).
2. DAMA's Data Management Body of Knowledge (DMBOK).
3. Federal Data Architecture Subcommittee (DAS) Data Quality Framework, v1.0, October 2008.
4. National Institute of Standards and Technology (NIST), Information Security Handbook: A Guide for Managers Information Security, Special Publication (SP) 800-100, Revision 3 (March 2007).

Definition: *Radio Frequency Spectrum Management is the analytical, procedural, and policy approach to planning and managing the use of the electromagnetic spectrum.*

Keywords: *harmful interference, policies and procedures, radio frequencies, radio frequency interference analysis, radio spectrum, system acquisition*

ENTERPRISE TECHNOLOGY, INFORMATION, AND INFRASTRUCTURE

Radio Frequency Spectrum Management

MITRE SE Roles and Expectations: MITRE's customers are becoming increasingly dependent on wireless communications, navigation, and surveillance systems in order to support a broad variety of operational missions in the areas of air traffic control, national defense, and homeland security. The single most critical asset that any wireless system must acquire is the radio frequency (RF) spectrum in which to operate. Nearly everywhere in the world, unallocated radio spectrum has become scarce, and as a result, its commercial value has increased dramatically. In the resulting intense competition for a limited resource, private companies have been winning the "war of words" associated with this asset. This makes it increasingly difficult for government agencies to acquire spectrum for new systems and even to keep the frequencies they have been using for years.

MITRE SEs are being called on to advise government system developers, operational units, and policy organizations on how best to plan for, acquire, use, and retain radio frequencies. It is essential for MITRE staff involved in systems that depend on RF emissions to have a working knowledge of this somewhat complex field and to be able to get help from MITRE experts when needed.

Government Interest and Use

All useful regions of the radio frequency spectrum (9kHz–300GHz) are regulated. Worldwide, the International Telecommunication Union (ITU), an entity within the United Nations, maintains a Table of Allocations to which most countries adhere, to a large extent [1]. The ITU has divided the world into three regions, each often having different radio rules and allocations. Each nation also has internal spectrum regulators who manage what is universally considered to be a sovereign asset within their own borders. Generally a Ministry of Telecommunications or similar organization fills this role.

The ITU is the venue in which deliberations are held to accommodate new types of telecommunications functions. World Radiocommunication Conferences (WRCs) are held every three or four years to consider changes to the Table of Allocations. Because this process takes several years to complete, spectrum for any new function (e.g., when satellites were first introduced in significant numbers in the 1970s) has to be planned for *many years* in advance.

In the United States, the authority to regulate spectrum use is split between two agencies: the National Telecommunications and Information Administration (NTIA) [2] and the Federal Communications Commission (FCC) [3]. The operating rules of these agencies are extensive and are codified into law within Title 47 of the U.S. Code of Federal Regulations.

NTIA is responsible for spectrum matters that involve federal government users in all three branches of the government. For a new system, the procuring federal government agency must provide the system's technical characteristics and demonstrate to the satisfaction of NTIA that the system neither causes nor receives harmful interference to or from other authorized users when placed in its intended operational environment. Once this is accomplished, NTIA issues a Certificate of Spectrum Support, which identifies the frequency band in which the agency can operate and bounds the technical parameters that the system can have. NTIA then issues a frequency authorization allowing the user to operate a system on a specific frequency or frequencies at a particular location or within a defined area. Once a system is fielded, a multitude of radio frequency analysis and spectrum management tools are available to plan for and identify frequency assignments. Ultimate authority, however, to use a frequency must come through an NTIA frequency authorization or through delegated authority, which is provided by NTIA to specified federal government agencies for certain bands.

The FCC is responsible for the spectrum matters of private users as well as state and local government users. The FCC first issues a Type Acceptance for new non-government systems, identifying the authorized frequency band and parameter set. For most systems, the FCC then issues a radio license that grants a user the right to use a particular frequency or range of frequencies at a given site.

It is worth noting that this bifurcated approval process can both complicate and protract the system acquisition process for MITRE's government customers. For example, to develop and test a spectrum-dependent system, a private sector vendor must follow the FCC's rules for doing so—even if the eventual end user is a government agency. The acquiring government agency must then go to NTIA to obtain the necessary approvals to use the system in an operational environment.

Best Practices and Lessons Learned

Know the spectrum policy landscape (part 1). The management—and very often even the technical staff—of most government system acquisition programs is not acquainted with the requirements, policies, and procedures associated with the identification, acquisition, and retention of adequate radio spectrum resources for their systems.

Know the spectrum policy landscape (part 2). MITRE SEs involved with spectrum-dependent systems should have at least a rudimentary understanding of domestic (NTIA and FCC rules) and international spectrum regulations and policy. MITRE SEs supporting the Department of Defense (DoD) should additionally be familiar with DoD Instruction (DODI) 4650.01, "Policy and Procedures for Management and Use of the Electromagnetic Spectrum" [4].

Know the planning horizon (part 1). The time required to obtain spectrum for a new type of system is measured in years. Typically, it takes six to ten years to get new spectrum to the point where systems can actually use it. The Office of Management and Budget requires that federal government agencies obtain an NTIA Certificate of Spectrum Support before submitting budget requests for "the development or procurement of major communications-electronics systems (including all systems employing space satellite techniques)." It is thus vitally important to initiate the processes to obtain spectrum for new system programs as soon as possible.

Know the planning horizon (part 2). Even if a new system does not represent a new radio service (e.g., a communication, navigation, or surveillance), it can take more than a year to obtain the approvals to use existing spectrum.

Dual approvals needed. Government contractors must follow FCC rules [3] for spectrum use during their design, test, and acceptance phases.

The acquiring agency must then get a separate (NTIA) approval to use the system on government frequencies.

Know the competition. Competition for radio spectrum has intensified in recent years, particularly in bands that are optimal for mobile systems (approximately 200MHz–4GHz). This factor has had a dramatic impact on the perceived (and actual) value of spectrum and has biased decisions for spectrum re-allocation heavily in favor of the private sector.

Importance of justification. Government agencies must develop compelling, operationally based justifications for both the acquisition of new spectrum and the retention of spectrum they already have. Failure to do so will cause spectrum to be lost to commercial interests with the resulting harmful impact on the mission of the federal government.

Design and architecture implications. Government agencies typically operate systems over long life cycles (e.g., 15–30 years or more). With growing scarcity of unused spectrum and rapid changes in technology, system designs should consider wider tuning ranges and modular architectures that facilitate upgrading over the life cycle. Such considerations are especially important for systems to be operated overseas in order to maximize the likelihood that the applicable host nation(s) will authorize such equipment to operate.

Leverage the corporation's expertise. MITRE has a strong capability in spectrum management that can be brought to the aid of systems engineers who are working with spectrum-dependent systems. As one entry point into the MITRE spectrum community, Dr. Chris Hegarty currently serves as MITRE's corporate focal point for spectrum. [3]

Share your information. MITRE SEs should inform the corporate focal point for spectrum, in addition to their own management chain, of any spectrum-related issues that involve more than one of our sponsors.

References and Resources

1. ITU Radio Regulations are available for purchase at http://www.itu.int/pub/R-REG-RR/en.
2. NTIA's Manual of Regulations and Procedures for Federal Radio Frequency Management, www.ntia.doc.org.
3. The FCC Rules are available on the FCC's website, http://www.fcc.gov/rulemaking.
4. Policy and Procedures for Management and Use of the Electromagnetic Spectrum. DoD Instruction 4650.01, January 9, 2009.

Additional References and Resources

Through an ongoing, corporately funded initiative, MITRE has developed and maintains a spectrum management Web collection intended to provide MITRE staff with a basic overview of spectrum management. Links are provided to domestic and international regulatory documents and websites. The site also includes a listing of MITRE documents related to spectrum management, points of contact within MITRE's staff, and directions for joining the MITRE spectrum shared user distribution list (which currently has over 60 members).

Engineering Information-Intensive Enterprises

Definition: *An enterprise is a network of interdependent people, processes, and supporting technology not fully under the control of a single entity. Successful operation of an information-intensive enterprise substantially depends on networked information systems. Engineering an information-intensive enterprise concentrates on managing uncertainty and interdependence in an enterprise; involves engineering both the enterprise and the systems that enable it; and is directed toward building effective and efficient networks of individual systems to meet the objectives of the whole enterprise.*

Keywords: *architecture, change, composable, design patterns, information-intensive, innovation, mission assurance, open systems, uncertainty*

Context

The success of our sponsors' organizations increasingly relies on information. If the right information isn't available when needed, the missions and outcomes of the enterprise will be less effective, efficient, or successful.

An enterprise has many components and information that must come together for mission success. Data, business rules, applications, communications, and sensors need to be created or composed into capabilities within the constraints of the enterprise's architecture(s), designs, existing systems, and mission assurance requirements. Here are a few examples:

- For homeland security, communications capabilities must support the needs of first responders and state, local, and tribal partners.
- In the DoD, cyber security threats require careful consideration and close examination of open capabilities and emerging technologies, such as social networking, before employing them.
- In air traffic management, the need for public trust may drive the business rules associated with free flight and use of unmanned systems in the national airspace.
- For modernization efforts like those at IRS and VA, questions arise about how and when to insert new technology and capabilities in light of the readiness of the operational organizations to absorb them and their associated new processes and procedures.

Articles Under This Topic

Articles under this topic are intended to help MITRE staff in engineering information-intensive enterprises.

Architectures are used by and across our customers for a variety of purposes—to support understanding of operations, help with system design and implementation, and provide basic building blocks for enterprise capabilities. A federated architecture helps deal with the magnitude and complexity of engineering cross-enterprise needs to enhance overall mission effectiveness. The article "Architectures Federation" discusses how federated architectures enable local innovation, enterprise integration, and evolution across major portions of an enterprise—many of which may be enterprises in their own right.

Design patterns in software are not concrete pieces of software, but a kind of stencil of best practices applied in certain situations. MITRE systems engineers (SEs) are likely to encounter and use them in developing or reviewing interfaces between system components or, at a higher level, across system boundaries. The article "Design Patterns" describes basic approaches, best practices, and lessons learned in using these patterns in engineering service-oriented environments and interface standardization activities.

The article "Composable Capabilities On Demand (CCOD)" describes a new and evolving strategy to enable the rapid piecing together of capabilities to meet end users' needs, in some cases by the users themselves. CCOD is in the style of many Internet tools that enable the rapid application of various services to data or information to compose a "user-defined" or tailored view/perspective to satisfy their needs.

Open systems approaches enhance the ability to rapidly create capabilities in information-intensive systems. The article "Open Source Software (OSS)" provides an historical perspective on OSS, describes the rapidly changing view of OSS and its relationship to engineering information-intensive enterprises, highlights government interest in and use of OSS, and concludes with a comprehensive and detailed set of best practices and lessons learned in applying open system techniques and using open source software.

MITRE systems engineers should understand the legal requirements that apply to federal agencies' collection, use, maintenance, and disclosure of personally identifiable information. The article "Privacy Systems Engineering" provides guidance on how privacy must be built into the systems engineering life cycle and how technology can be leveraged to protect privacy.

MITRE SE Roles and Expectations

MITRE systems engineers are expected to develop enterprise solutions that balance local innovation with global innovation and evolution. They develop solutions that (a) provide customized innovations to meet end-user local needs, and (b) interoperate with, respond to, and co-evolve with an environment that itself is constantly changing.

Best Practices and Lessons Learned

Information as capital. Treat enterprise data and information as a capital resource that has value over time. Emphasize the importance of a data strategy in your work.

Data interoperability. Adopt the view that data interoperability should be engineered to ensure that cross-enterprise capabilities are realized.

Be attuned to enterprise cycles. There are long- and short-term customer cycles. The former includes activities like budgeting, requirements, contracting, and implementing. The latter includes responding to urgent operational needs. Understand and differentiate between them, and adapt systems engineering to them.

Consider capability longevity. Understand the likely longevity of the capabilities that users need. Adapt your perspective and systems engineering approach to this aspect of the capabilities you engineer. A capability might be required for the immediate situation/environment, but then not be needed for the next crisis, or ever again. In a crisis, consideration of capability evolution might not be a critical part of the systems engineering analysis, but consideration of future use should not be completely set aside. For example, a design pattern could be used to create an immediate capability that, at the same time, facilitates use for future crises. A composable capability strategy can enable components to be created and be "on the shelf" to support future situations. Open source capabilities can provide a foundation for "immediate use."

Don't throw away "throwaway" thinking. Many customer developments stress that everything must be able to be reused by others (and this has intensified in the service-oriented world). Although this is often the case, sometimes the prudent course of action is to build a faster, cheaper, throwaway capability. Understand the value of reuse within your enterprise and by others, but also understand that in some situations building a throwaway version is the better course of action.

ENGINEERING INFORMATION–INTENSIVE ENTERPRISES

Architectures Federation

Definition: *Architecture federation is a framework for enterprise architecture development, maintenance, and use that aligns, locates, and links separate but related architectures and architecture information to deliver a seamless outward appearance to users.*

Keywords: *enterprise architecture, federated architecture, fit for federation, semantic alignment, tiered accountability, touch point*

MITRE SE Roles and Expectations: MITRE works with a variety of government customers to help them build enterprise architectures, often in the context of supporting their overall enterprise modernization or transformation programs. Many customers are facing the complex problem of sharing their business processes, information stores, technical systems, and human resources in a cohesive and secure way to accomplish a common mission. MITRE systems engineers (SEs) are expected to understand and apply the principles of architectures federation to enable local innovation, enterprise integration, and evolution across major portions of an enterprise architecture or multi-agency enterprise architectures. By helping them build their respective products to meet common prescriptive direction, MITRE's customers will be able to reuse component architectures by "snapping them together" like LEGO® bricks to build complex architectures of wider scope and applicability.

Introduction

In recent years, MITRE has been supporting architecture efforts across the federal government spectrum. In fact, the federal government now mandates the use of Enterprise Architectures (EAs) by agencies seeking to obtain funding for any significant information technology investment. Customers use architectures to improve warfighting and business capabilities by enhancing the interoperability and integration of U.S. enterprises (e.g., the Air Force Enterprise) with Joint and Coalition forces, other Services, and national agencies.

To accomplish the preceding efforts, MITRE SEs are expected to understand and apply the principles of federated architectures to account for architecture interrelationships and to express how architectures connect to one another. Federated architectures enable local innovation, enterprise integration, and evolution across major portions of an enterprise—many of which may be enterprises in their own right. Principles of architectures federation in practice require merging, integrating, and federating a large number of diverse organization architectures such as the Federal Aviation Administration, DoD, DHS, CBP, and the Federal Emergency Management Agency, as well as contributions from industry players like the airlines, airports, IT industry, weather bureaus, and others. This article explores the basic concepts of architectures federation and offers lessons learned to help MITRE systems engineers understand how the principles of federation can help practitioners build architectures more efficiently and effectively.

What is enterprise architecture?

Architecture relates to the structure of components, their relationships to each other and to the environment, and the principles guiding the design and evolution of the entity they describe [1], whether that entity is an organization (e.g., federal department or agency), a system (e.g., Joint Surveillance Target Attack Radar System), or a functional or mission area (e.g., financial management, homeland security). Architecture products and artifacts can take a variety of forms, including models of structured data stored in an architecture tool or database repository, graphical depictions of the information in hard copy or electronic format, or unstructured data or text.

A good working definition of "enterprise" is any organization or group of organizations that has a common set of goals or principles, or a single bottom line (e.g., a corporation, a single department, a government entity, a network of geographically remote organizations). An enterprise architecture provides a clear and comprehensive picture of an enterprise. It consists of snapshots of the current operational and technological environment, the target environment, and a capital investment roadmap for transitioning from the "as is" to the "to be" environment. In other words, it acts as a roadmap for the way ahead. The snapshots are further comprised of "views," each of which consists of one or more architecture products that provide conceptual or logical representations of some part of the enterprise of interest to a particular group of stakeholders [2].

What does federated architecture mean?

The historical approach of developing monolithic, integrated architectures has not worked well, as these products generally become too complex and unwieldy. By contrast, a federated architecture is a framework for enterprise architecture development, maintenance, and use that aligns, locates, and links separate but related architectures and architecture information to deliver a seamless outward appearance to users. It enables a complex architecture to be built in a piecemeal fashion from component architectures. In this way, a federated architecture approach recognizes the uniqueness and specific purpose of individual architectures, and allows for their autonomy and local governance, while enabling the enterprise to benefit from their collective content.

Federation provides the means to organize an enterprise's body of knowledge (architecture) about its activities (processes), people, and things within a defined context and current/future environment. Federated architectures support decision making by linking architectures across the enterprise, providing a holistic enterprise view that allows for the assessment of such matters as interoperability, identification of duplication and gaps, and determination of reusability [1].

Why develop architectures that support federation?

The ability to integrate and/or federate architectures is essential for addressing enterprise issues across a broad domain such as a federal department or agency. Federation

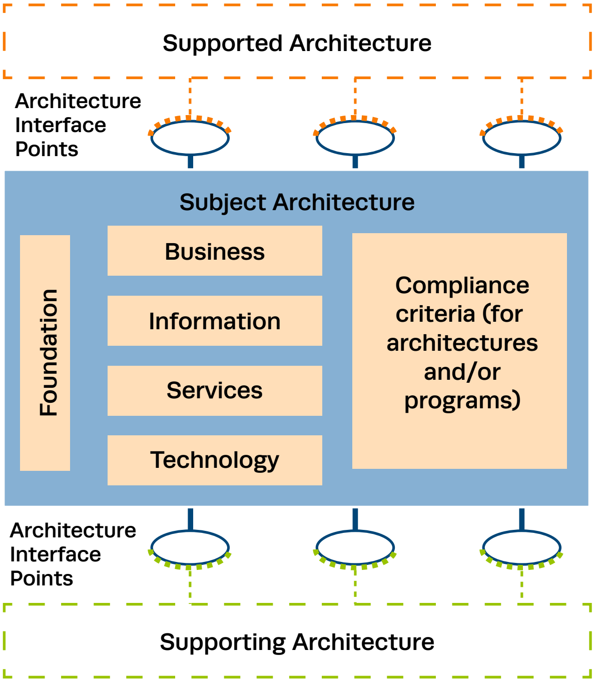

Supported Architecture
Architecture Interface Points
Subject Architecture
Foundation
Business
Information
Services
Technology
Compliance criteria (for architectures and/or programs)
Architecture Interface Points
Supporting Architecture

Figure 1. Key Constructs for Architectures Federation

enables multiple groups to develop architectures with the focus that best meets their immediate needs, while providing a means for linking and relating those architectures to address issues that cross multiple areas. A single architecture may not be able to address the entire enterprise sufficiently to support the kind of analyses needed in a large organization with a diversity of missions. The ability to federate multiple architectures leads to a more robust construct for understanding the enterprise in smaller, bite-size chunks.

Architecture federation serves, in part, as a process for relating subordinate and parent architectures via finding overlaps and establishing mappings between their common architecture information. Federal departments and agencies are also pursuing another use of an architectures federation strategy that divides the enterprise into manageable, right-sized components, each of which can be described by the communities that are most closely associated with them [3]. A small set of rules, common terms, and standards are used by everyone to maintain consistency so that the component parts can be "snapped together" as needed. For example, department architectures depict department-wide rules and constraints, component architectures depict mission-specific services and capabilities, and solution architectures depict solutions that conform to higher rules and constraints.

The concept of federation also plays an important role in the development of the environment and the sharing of information. For example, as federal department and agency enterprises become increasingly networked, federated architectures are proving essential in organizing the array of information and complex relationships. Federated architecture metadata is also useful for evaluating portfolios of existing systems and programs to make decisions about changes or additions necessary to achieve desired capabilities.

So then, what is federated enterprise architecture?

As defined by the enterprise scope, federated enterprise architecture is a collective set of architectures with the following attributes:

- It operates collaboratively, where governance is divided between a central authority and constituent units, balancing organizational autonomy with enterprise needs.
- The central authority's architecture can focus on the dynamics of economies of scale, standards, and the well-being of the enterprise.
- Constituent units' architectures have the flexibility to pursue autonomous strategies and independent processes [4].

What are the central elements that support architectures federation?

In a federated approach, responsibility for architecture development is shared at different echelons within the enterprise. To bring these separate but related efforts together requires:

- **Tiered accountability:** Establish a hierarchy of architectures whereby architectures lower in the hierarchy inherit characteristics from higher-level architectures. Use touch points to relate architectures across the levels or tiers.
- **Categorization:** Relate and group "like" architectures and artifacts.
- **Semantic alignment:** Use common vocabulary and mapping relationships to establish shared understanding.
- **Reference architectures:** Provide parent taxonomies for other architectures to use.
- **Search and discovery:** Allow authorized users to find and access relevant architecture for information and reuse [3].

What are some key constructs for architectures federation?

The key constructs for architectures federation are graphically depicted in Figure 1. Each construct comprises a collection of architecture products of interest to a particular group of stakeholders.

The subject architecture is the architecture that drives solutions for a specific purpose. It addresses all the business, information, business services, and technology components needed to deliver capabilities. The architectures of those solutions upon which the subject architecture relies are called supporting architectures, whereas the architectures of those solutions that rely on the subject architecture are called supported architectures.

Each architecture interface point (also called touch point) is an abstract representation of a purposeful connection between two architectures. These architecture interface points are abstractions of real-world interfaces that will be embodied in the solutions that implement the corresponding architectures. In simple terms, the interface points are the places where architectures can be joined into a larger federated architecture, so they are key to purposeful federation from an operational perspective [5].

What is the role of compliance in federation?

It is important for an architecture to comply with a set of standards, if it will be shared and used to support federation with other architectures (e.g., guiding the development of other architectures or programs). These standards come in the form of prescriptive direction called compliance criteria. Compliance criteria include business rules and processes such as information, service, and technology standards. A program or other architecture must adhere to these for it to comply with a given structure. Compliance criteria are augmented with descriptions of the ways in which these criteria will be verified. Therefore, the compliance criteria explicitly state what a program or architecture must demonstrate in terms of functionality and in terms of adhering to standards and meeting specific qualitative requirements.

An organization can start by creating architectures that meet a minimum set of standards, making it easier to share the architectures and positioning them for use in building a federation of architectures to support the construction of a federation of interoperable solutions.

What are some examples of compliance criteria?

Fit for Federation is an example of a specific compliance assessment that might be applied to any architecture that will become part of an architectures federation. Fit for Federation is determined by the following compliance criteria:

- The architecture's purpose has been documented and verified by users and usages.
- Input has been verified as coming from authoritative source, and the authoritative source is recorded.
- The architecture and/or analysis (output) have been verified as fit for purpose.
- Supported architecture interface points and associated standards are identified, documented, and verified.
- Supporting architecture interface points are identified, documented, and negotiated with the provider.
- Other compliance criteria (e.g., enterprise-wide standards and/or qualitative requirements) are established, documented, and verified.

Some examples of qualitative requirements that might be applied while assessing conformance to compliance criteria are affordability, dependability, extensibility, performance, and trust.

For a service-oriented environment, specific compliance criteria would be packaged as service-level agreements (SLAs). A single compliance criterion can distribute to multiple SLAs. For example, supporting a given vocabulary would apply to all services that deal with the subject (domain) vocabulary.

Lessons Learned

To federate architectures, there must be semantic agreement so that pertinent information can be related appropriately. MITRE SEs can recommend that their customers achieve semantic agreement by:

- Adhering to a common framework, which includes the use of common data element definitions, semantics, and data structures for all architecture description entities or objects
- Conforming to common or shared architecture standards
- Using enterprise taxonomies and authoritative reference data.

In general, conforming to common or shared architecture standards increases interoperability and makes it easier to federate. MITRE SEs should encourage their customers to choose standards appropriate to their purposes and help them establish the means to enforce compliance. For example, agreed enterprise taxonomies establish the context for aligning mission-area activities and associated reference models, and for categorizing and organizing component architectures, thereby facilitating semantic understanding across the various architectures in the federation.

The federation of architectures is facilitated by an environment that enables information sharing. MITRE systems engineers first must recognize that an architecture-sharing environment requires sound governance and enterprise architecture services. They must help their customers establish sound governance structures to apply accountability to the development and maintenance of architectures toward set objectives, which will ultimately facilitate their ability to federate. This approach places responsibility around processes such as configuration management and quality assurance. MITRE SEs also must encourage their customers to establish enterprise architecture services to allow for the visibility, accessibility, and understandability of architecture information in a consistent and efficient manner.

The success of a federation effort also depends on exposing architectures and architecture metadata for potential linkage and reuse by analysts, planners, and decision makers at every level. Sharing architectures and services that already exist helps expedite architecture development and federation. Registry capabilities [6] provide for registration and linking of architecture metadata to enable the creation of navigable and searchable federated enterprise architectures. Enterprise enforcement policies and governance for architectures reinforce robust interfaces and data relationships [1]. MITRE systems engineers should assist their customers to actively engage in these architecture-sharing venues by reusing artifacts before reinventing them and by posting their own metadata and products for reuse by others.

MITRE SEs should promote and foster the development of federated architectures within customer organizations to help improve the reliability and efficiency of decisions. This will occur as organizations align semantic and structural data across their boundaries so they can ensure that the right information is being used to answer key decision makers' questions. MITRE systems engineers should continue to use federated architecture opportunities and improve the flow of information among stakeholder nodes and consequently decision makers.

Summary

MITRE is working with a wide variety of government customers to help them build their EAs, most often in the context of supporting their overall enterprise modernization or transformation programs. A key skill that MITRE systems engineers need to bring is an understanding of how business needs, information technology, and people come together in well-constructed architectures.

Many of MITRE's customers are facing the complex problem of multi-agency enterprise architecture. How can different government entities share their business processes, information stores, technical systems, and human resources in a cohesive, secure way to accomplish a common mission? Architectures federation can foster this kind of sharing. By helping them to build their respective products to meet common prescriptive direction, MITRE's customers will be able to reuse component architectures by "snapping them together" like LEGO® bricks to build complex architectures of wider scope and applicability.

References and Resources

1. Department of Defense, April 23, 2007, *DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines*.
2. Hite, R. C., and G. D. Kutz, March 28, 2003, *DoD's Draft Architecture*, GAO-03-571R.
3. Frey, B., July-September 2008, "Department of the Navy Architecture Federation Pilot," *CHIPS*, pp. 41–43.
4. Air Force Chief Architect's Office, December 2007, *Air Force Architecture Framework*.
5. COLAB—Collaborative Work Environment, http://colab.cim3.net, accessed January 20, 2010.
6. Department of Defense, DoD Architecture Registry System, accessed January 20, 2010.

Additional References and Resources

Business Transformation Agency, BEA 6.2 Informational Release, http://dcmo.defense.gov/products-and-services/business-enterprise-architecture/

Federal Chief Information Officer Council, February 2001, A Practical Guide to Federal Enterprise Architecture, Ver. 1.0.

"System Architecture," Project Leadership Handbook, The MITRE Corporation.

Definition: *Design patterns in software are usually short descriptions capturing practices that have proven successful in the past. They are not concrete pieces of software, but a stencil applied in certain situations. They are generally not prescriptive, but suggestive; include guidance on their most appropriate use; and provide examples from existing systems. Their most important use is to describe the interaction of objects or systems with their environment (i.e., other objects or systems). Design patterns can occur at different levels of system design, from low-level programming to system-of-systems. At the latter level, they are most associated with interface design and coupling.*

Keywords: *coupling, design pattern, interface*

ENGINEERING INFORMATION-INTENSIVE ENTERPRISES

Design Patterns

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the general principles and best practices of design patterns for information technology (IT) intensive systems. They are expected to select and recommend the patterns appropriate to the application, understand the challenges and choices that arise, and understand the issues and challenges of interface design in an enterprise environment.

Background

The concept of design patterns is usually attributed to the work of the architect Christopher Alexander, and was adapted to software by Kent Beck and Ward Cunningham. In 1995, the popular book *Gang of Four* (GOF) [1] established a set of patterns that are in continuous use, and provided a "pattern" for describing the patterns. These 23 patterns are divided into creational, structural, and behavioral categories. Many other patterns have been defined, as well as other categories, such as user interface.

As an example, one GOF patterns is the *Abstract Factory*, a creational pattern that presents an interface for creating new objects, without the caller knowing the specific type of object being created. This could be used to implement a different look and feel with minimal changes to the program. Other examples are the *Proxy* structural pattern, in which one object becomes a surrogate for another (with the same interface), often used in remote procedure calls, the *Singleton* pattern, in which a class allows only one instance of itself to be created, often used in managing shared resources, and the *Mediator* behavioral pattern, which allows loose coupling between classes by being the only class that has detailed knowledge of their methods.

Design patterns enable review and discussion of software design to take place at a higher and more abstract level than reviewing specifics of interface calls. We can ask: "Should you be using a *Singleton* pattern here?" or "Would an *Abstract Factory* pattern help?"

GOF patterns have several things in common: they are defined in terms of object-oriented software, they (usually) describe the interaction of an object with its environment (e.g., other objects), and they are generally used within the internal design of a single application (i.e., a local calling environment).

Patterns can also be viewed at a broader level of design, however, and MITRE SEs are more often involved in this aspect. MITRE SEs are less likely to be involved in the development of the detailed internal workings of system components than in the review of interfaces between components or, at a higher level, between systems. This calls for a set of design patterns that focus on the manner in which connections are made across the system boundaries. Many GOF patterns will not directly apply.

Design Patterns in an Enterprise Engineering Service–Oriented Environment

Two considerations arise when designing for a large-scale enterprise service environment: (1) users may put services, interfaces, etc., together in ways that designers did not anticipate, and (2) any interface changes will affect a larger set of users. Thoughtful use of design patterns can help deal with both of these issues. A third issue with scaling to the enterprise is that a service will generally have to deal with a (currently) unknown and potentially large number of users. Design patterns are of less use in dealing directly with this issue.

In an enterprise environment, when considering system-to-system interfaces, the notion of design patterns can be broadened to encompass more general guidance on how to manage the coupling in the interface. As a general rule, loose coupling is preferred over tight coupling whenever possible. Loose coupling means that a change in the implementation of one side of the interface does not affect the implementation of the other side. For example, using a code in a field with a lookup table that must be distributed to users is not loose coupling. Also, a loosely coupled interface should not lock in specific limits that will inhibit scalability. As a simple example of this, in an interface for contact information, allowing for only one (or two) telephone numbers of 10 digits may not be sufficient. A more extensible interface might allow for an arbitrary-length list of telephone numbers of indeterminate length.

Loose coupling insulates users of an interface from changes in the implementation. For example, a well-designed interface should be able to add additional parameters to the interface, while still generating and accepting messages without the new parameters. This allows for growth and innovation without stranding users of the previous version of the interface. On the flip side, though, this extension mechanism must be managed with discretion, or the number of supported interfaces that differ just in parameters can grow large, and the maintenance of these can swamp the value of backward compatibility.

Interface Standardization Efforts

Cursor on Target (CoT) is an example of an enterprise effort to simplify a collection of interfaces and provide loose coupling. The Air Force has had a large number of tightly coupled point-to-point interfaces among many components. Gen. Jumper (former Chief of Staff of the Air Force) inspired MITRE to come up with a small set of data elements that would give the majority of what most users need. MITRE studied several months' worth of messages and found that a small number of data elements were used repeatedly. CoT standardized a definition of these elements in an XML format that is easy to generate and parse. It provided for compatible extensions so that new elements could be added without disrupting existing users.

Universal Core [2] (UCORE), developed by the Department of Defense (DoD) and the Intelligence Community, built on the CoT philosophy and approach. It is hierarchically designed to allow the user to choose the level of detail desired in a particular element. Users can find out that an object is a fixed wing aircraft, or drill down and find out the type of aircraft (e.g., F16), or even a unique aircraft identifier such as the tail number. This pattern helps to define data elements that are common across multiple communities of interest. It follows several principles:

- Be able to operate at different levels depending on the needs of the user (hierarchical).
- Make schemas extensible.
- Develop small spirals, making it easier to build innovations.

Alignment with MITRE Systems Engineering Competency Model (SE CM)

Systems engineering work with design patterns most closely aligns with the "Architecture" (Section 2.3) and "Software and Information Engineering" (Section 4.7) competencies in the MITRE SE CM [3]. In the former, design patterns can be a useful tool in discussing, visualizing, comparing, and recording architectural interface decisions. In the latter, because design patterns are now a well-established paradigm within software engineering, an understanding of the techniques and terminology is useful in facilitating communication between the customer/user and software specialist.

Best Practices and Lessons Learned

The following rules of practice can be seen as design patterns for interfaces at the enterprise level as well as at the detailed implementation level:

Avoid complexity in the interfaces. Complex interfaces typically do not scale well. The complexity is pushed out to all users, and the skill in dealing with it may vary. For example, rather than providing latitude and longitude in 10 potentially different formats, each of which has to be handled by the user, provide it in a single format instead. If an interface is too complicated, there is a greater possibility that it will be misinterpreted, or that developers will copy sub-optimal implementations of the user end. Complexity leads to errors, which can lead to poor performance that may not be correctable, and may even become security risks.

Use loosely coupled interfaces wherever possible. Loose coupling implies that a change in the implementation of one side of the interface will not affect the implementation of the other side. This allows enormous freedom on both sides to make improvements and to keep development schedules disjoint. Tight timing requirements or (unfortunately) software version requirements may be considerations that require a reevaluation and relaxation of this practice, but this should be made explicit and documented in such cases.

Use tightly coupled interfaces only if they are necessary for performance. Tight coupling can lead to code that is buggy and fragile. An example of tight coupling is in the Link-16 interface, which, because it is a tactical link, uses a number to represent the type of an aircraft. This ties the user to a particular version of a conversion table. If the table is updated on one side, the user may be left with a meaningless number until the table is updated as well. Of course, a more expansive communication protocol could carry all information on the aircraft explicitly, but bandwidth limitations may prohibit this as an alternative.

When possible start design with loose coupling. Even in cases where tight coupling will be used, initial design can begin with loose coupling interfaces. Document why a tight coupling is being used. This is analogous to defining a logical schema in a database management system (DBMS)-independent way, but implementing it in a

DBMS-dependent physical schema. This may be a useful pattern for systems of systems.

Focus on data conformity in the interfaces rather than in internal representations. In the 1990s, government organizations tried to enforce data uniformity across all applications, even to the point of specifying how data was to be represented within the application and its databases. This was never achieved. More recently, the focus is on creating common definitions for data exchange, leaving applications free to choose how to represent data internally [4, 5]. This has proven to be an easier goal to reach.

Recognize that differences in the representation of data result from different uses of the data. For example, consider a gun. A shooter wants to know its range, caliber, etc. A shipper wants to know its size, weight, etc. Finance wants to know its cost, estimated lifetime, etc. The same gun is naturally represented differently in different systems. Forcing all characteristics on all systems would be burdensome. However, unanticipated, innovative uses of data can be achieved through compositional patterns to create new data representations that are built on existing representations.

In the design of an interface, consider the 80/20 rule. It may be better to implement 80 percent (or so) of what most users need most of the time, especially if this can be done quickly with a simple interface. This reduces the cost and time for implementation.

Build in the ability to extend the interface. Some users will need to reach at least part of that remaining 20 percent, and in any case, interfaces have to grow and change over time. A loosely coupled interface should build in a mechanism for compatible extension, so that changes and additions can be made without affecting users who do not need the extensions.

Consider the governance of the extensible interfaces. Extension of an interface creates multiple versions/copies that must be managed. Consider the justification for and understand the impact of doing this.

Do not forget about the semantic level of understanding in the interface. It is fine for someone to be able to correctly parse your interface, but there must also be agreement on the meanings of the data elements.

Involve developers in the development of system interfaces. Those who will implement the interface should be involved in the design, since they may have insight into decisions that could inhibit scalability or cause other problems.

References and Resources

1. Gamma, E., R. Helm, R. Johnson, and J. Vlissides, 1995, *Design Patterns—Elements of Reusable Object-Oriented Software*, Addison-Wesley.
2. November 2008, UCore: Breaking the Barrier to Information Sharing.
3. The MITRE Corporation, The MITRE Systems Engineering Competency Model.

4. Department of Defense, September 26, 1991 (certified current as of November 21, 2003), DoD Data Administration, DoD Directive 8320.1.
5. Department of Defense, December 2, 2004 (certified current as of April 23, 2007), Data Sharing in a Net-Centric Department of Defense, DoD Directive 8320.02.

Additional References and Resources

"Agile Acquisition," MITRE Project Leadership Handbook, The MITRE Corporation.

Erl, T., 2009, *SOA Design Patterns*, Prentice Hall.

Fowler, M., 2002, *Patterns of Enterprise Application Architecture*, Addison-Wesley.

Hohpe, G. and B. Woolf, 2003, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley.

National Information Exchange Model, Accessed September 16, 2009.

"Prototyping/Experimentation/Modeling and Simulation," MITRE Project Leadership Handbook, The MITRE Corporation.

"System Architecture," MITRE Project Leadership Handbook, The MITRE Corporation.

Definition: *Composable Capabilities On Demand (CCOD) is a design concept to enable the rapid development of new capabilities, carried out by operators combining services, data, and existing systems to achieve awareness of, or respond to, a new situation or mission.*

Keywords: *capabilities, components, composability, net centric operations (NCO), net-centric waveform (NCW), reuse, service*

ENGINEERING INFORMATION-INTENSIVE ENTERPRISES

Composable Capabilities On Demand (CCOD)

MITRE SE Roles and Expectations: CCOD is a new and evolving concept rooted in net-centric principles and effective enterprise-level distributed computing tenets (e.g., modularity, loose coupling, platform independence). CCOD disrupts traditional software systems engineering in two ways: the extension of capability composition to the *end user* as opposed to the developer; and enablement of the user to perform *runtime* composition of said capabilities. CCOD represents a dynamic composition of existing and emerging components; the result may even be a recombination of existing capabilities as opposed to a new system.

MITRE SEs attempting to apply CCOD principles must understand and use detailed expertise in these areas:

- Distributed and enterprise software engineering from low-level infrastructure to modular componentization
- Human systems integration for design, workflow analysis, and capability management, and acquisition, especially of malleable/componentized net-centric systems as opposed to large, monolithic, self-contained systems
- Security, information assurance, and mission assurance, which are especially challenging due to the run-time composition issues
- Governance, including contract models for component development/sustainment, infrastructure development/sustainment, testing, and related issues

Because CCOD is an evolving concept based on still-evolving technologies and tenets, MITRE systems engineers seeking to apply CCOD must be aware that not all programs will be amenable to such an approach.

Background

CCOD is a design concept to enable the rapid development of new capabilities by combining services, data, and existing systems to respond to a new situation or mission. Ideally, CCOD should enable operational users in the field to do this development. CJCSI 3170.01G, Joint Capabilities Integration and Development System, defines a capability as "*the ability to achieve a desired effect under specified standards and conditions through combinations of means and ways … . It is defined by an operational user and expressed in broad operational terms [1].*" CCOD supports this definition by providing an environment and associated components to enable the operational user to create relevant capabilities when needed.

Composable components may include elements from traditional services and data sources (e.g., government programs of record) as well as non-traditional ones (e.g., Twitter, Flickr, and other open-source/Web 2.0 technologies that foster the creation of user-generated content and capabilities). The resulting capabilities and combinations of services can be tailored to the task at hand and either published as a "template" (if useful to others) or discarded. Component pieces can be reused should a similar need arise in a future task.

Interoperability is facilitated by the use of "loose couplers"—common data formats that leverage the information-sharing benefits of the "network" through the exchange of sufficient mission data for the most common use. Loose couplers (e.g., Keyhole Markup Language [KML] and Universal Core [UCore]) are defined and evolved by a stakeholder community, including operational users, developers, and MITRE domain or technology experts. Experience to date suggests that a minimalist approach to loose couplers is a key principle.

Global, complex data standards may prove problematic, due to the large investment in parsing/integrating/adopting them.

Ad-hoc composability figures prominently in this development process; consequently, it is important that the component technologies be built to high usability standards, and that tools, services, and data sources be described in a manner that improves discoverability. Eventually it may be possible for end users with no specific training to find, combine, and use tools, services, and data sources in a mission-appropriate way via approaches like social networking. The widespread adoption of the CCOD approach also depends on innovation in the development of tools, services, and data sources. Innovation is crucial for ensuring that there is a sufficient variety of components to allow for the development of a mission-focused solution. This innovation can be fostered by the adoption of a layered architectural approach, which promotes the creation of multiple, partially overlapping capabilities. This, in turn, allows non-programmer end users to select the best components to incorporate into their solution.

Examples of CCOD capabilities can be found in the Agile Capability Mashup Environment (ACME) [2]. This environment has supported a variety of customers by rapidly bringing together innovative ideas, approaches, and capabilities to meet challenging problems.

Operational Properties of a CCOD Environment

The objective of CCOD is to provide operational capability to the end user. To make this possible, capabilities in the CCOD environment have the following properties:

- **User-facing:** The composed capability will deliver an effect for the end user. This doesn't mean that capabilities will not have middleware components, but rather that middleware alone is not sufficient to provide CCOD.
- **Rapidly integratable:** Reusable components in a CCOD environment can be rapidly integrated, ideally by the end user.
- **Quickly adaptable to the task at hand:** Capabilities that are composed can be quickly adapted to changing operational context.
- **Assured:** The capabilities that are composed will need to be assured from a mission perspective; this involves issues such as information security, trust, versioning, and verification/validation.
- **Integratable with existing operational systems and domain business processes:** Components designed for CCOD will need to integrate with existing business processes, command and control systems, and ancillary information technology (IT) systems.

The use of CCOD in government systems development and the acquisition process is nascent. CCOD's run-time and user-facing tenets provide challenges that are rarely considered in the systems engineering and acquisition realms.

Best Practices and Lessons Learned

These lessons have been derived from hands-on CCOD prototyping and operator-centric analyses of current command and control (C2) systems and workarounds, filtered through the lens of practiced distributed computing expertise. However, remember that CCOD is still a budding concept and these lessons are still forming.

Favor the small and reusable. Using Occam's razor, where there is a choice between developing or using two or more components, usually the small, simpler one is preferred. A component should be "light weight" to ensure its ease of adoption and integration across a variety of users and uses.

Make components discoverable. Ad-hoc mission-focused composability necessitates the ability to find the components and data best suited for a particular task in timely manner. A "marketplace" or "app store" concept is a useful construct for many CCOD environments.

Develop components with an understanding of the end user. Early experience is leaning toward a CCOD design concept that follows a multilevel producer/consumer design pattern. An engineer with operational/domain knowledge will still develop/compose some components, but the promise of CCOD will be fulfilled when the user/operator composes new functionality from existing components. Throughout the composition and use process, several users have differing roles and responsibilities:

- **Combat coder or "mashup engineer":** Develops, prepares, and publishes the data and services for consumption. This engineer has operational/domain knowledge and can compose data and visua lizations into raw application components for users.
- **Average users/operators:** Tailor the raw application components to meet their specific needs, responsibilities, and preferences. This is the first layer of users who consume or interpret the data, potentially adding, modifying, or filtering it before sending it up the chain. This is a typical operator in a mission setting who has potentially complex, mission-centric responsibilities yet is not a computer programmer.
- **Commander:** High-level information consumer who combines data from several sources to make final decisions.

Focus on reuse. Perhaps the greatest value of composable capability is the reuse of someone else's components and compositions. Each CCOD component should be reusable and generic, allowing other CCOD projects to use it, particularly outside the direct composition environment. Each solution should be used by successive CCOD projects to build on previous experience and lessons learned. Where possible, use existing open source solutions/tools that have adoption momentum and are adequate to the task at hand (e.g., Restlet, PostgreSQL, PostGIS, Smile, Jetty).

Strongly consider RESTful architectures. This framework has proven to be robust and flexible enough to allow for quick development and integration of components. Consider the Web Application Description Language (WADL) data standard [3], which has been used to facilitate communication between RESTful services. While WADLs have some potential restrictions as a result of their simplicity, these restrictions were not a hindrance in a Department of Defense project using CCOD. In fact, it was an advantage to the implemented architecture. WADLs are intuitive, easy to write and understand, and require minimal effort for cross-service communication.

Strongly consider loose couplers. Design components and services to be independent of one another through use of loose couplers. In some MITRE CCOD projects, effective loose couplers proved to be:

- UCore for mission data interoperability [4]
- KML for geo-referenced data [5]
- WADL for RESTful services
- CoT for tactical mission data interoperability [6]

Use standard industry data formats as loose couplers for ease of reuse, integration, and adoption of components. Loose couplers can reduce the number of required data translations from N2 to 2N.

Explicitly design component granularity. Components of a composition must be at the appropriate abstraction level. For nontechnical users to perform composition, the components must be abstract enough to be understood, flexible, and unusable. Minimize interaction between the components by using a loose coupler. Document dependencies between components/services. It is important to test the component abstraction level with the intended user population (e.g., average user, combat coder, etc.).

Prepare design/artifacts and integration plan early to mitigate integration challenges. Due to the challenge of integrating diverse components at run time, it's important to develop system architecture and sequence diagrams in a CCOD project's early stages. This helps build a strong foundation when later creating an integrated system. Where possible, use approved design patterns to enhance extensibility and understanding. Clearly communicate the design and goal of the various project elements to the project team early in the process. And, have lower level component tests along the way to test each self-contained component, as well as enterprise-scope testing. Incorporate many iterations, with small increments to functionality, to enable testing.

Emphasize documentation. Documentation is crucial for any system implementing CCOD principles. The goal is to have combat coders and average users who may be unfamiliar with any component/service leverage or reuse this functionality. Document early on and throughout the development process, and provide easy discovery and navigation of the documentation. Agree on the documentation approach at the beginning of the project. As appropriate, use tools and formal modeling approaches to ease communication outside the project.

Separate visualization from function. One CCOD-based prototype developed at MITRE could not naturally conform to a typical Model-View-Controller (MVC) design pattern, yet separation of the visualization from data components remained critical. Data only has to be exposed in simple standards to be visualized in useful ways. These include Hypertext Markup Language (HTML) tables, Extensible Markup Language (XML) (plain, Real Simple Syndication [RSS], geoRSS, KML, RESTful Web services), comma-separated values, etc.

Accommodate dependencies. A CCOD system depends and relies more heavily on other systems' performance and availability. If known performance issues exist, you should duplicate data sources, where possible, so that no piece of the system completely depends on an unreliable component. Document dependencies using a modeling paradigm as appropriate.

Scope the use of CCOD. CCOD represents a very large solution space. It is important to have a well-defined use case for CCOD that can be scoped within given resource and time restrictions.

Seek operational context. Seek operational expertise early to define the operational scenarios and use cases to which a CCOD solution will be applied. Some use cases, situations, or entire systems may not be viable candidates for a CCOD approach (e.g., real-time semi-automated targeting systems).

Build from existing components. Where possible, CCOD will provide capabilities built out of existing components discoverable and integratable on the network and/or adaptors interfacing with "non-composable" systems of record. Seek ways to provide composable "pieces" of large systems.

Verification/validation. All components should be tested to ensure that they operate as designed, prior to being integrated into the component repository or storefront. Developer testing should be documented so that users of components will be able to better understand the approved use and limitations of components.

References and Resources

1. Chairman of the Joint Chiefs of Staff Instruction, March 1, 2009, CJCSI 3170.01G, Joint Capabilities Integration and Development System, accessed November 28, 2009.
2. "Agile Capability Mashup Environment (ACME) Lab Drives Innovation and Integration," sidebar in REACT Fine-Tunes Air Force Capabilities Before Live-Flying, accessed April 30, 2010.
3. W3C Member Submission, August 31, 2009, Web Application Description Language.

4. "Universal Core Advances Information Sharing Across Government Agencies," accessed April 30, 2010.
5. Open Geospatial Consortium (OGC), KML Standard, accessed February 9, 2010.
6. "Creating Standards for Multiway Data Sharing," accessed April 30, 2010.

Additional References and Resources

Alberts, D., J. Gartska, and F. Stein, July 2002, *Netcentric Warfare*, CCRP Publication Series.

ENGINEERING INFORMATION-INTENSIVE ENTERPRISES

Open Source Software (OSS)

Definition: *Open source software (OSS) is commercial software for which full ownership rights are obtained by agreeing (without immediate third-party verification) to abide by an attached OSS license. Agreeing to the license lets an individual, company, or government entity replicate, distribute, and run the application as often and as broadly as desired; obtain its human-readable source code; and (subject to release requirements that vary from license to license) expand or extend the OSS application. Payment is indirect, usually consisting of agreeing to share value (e.g., application fixes and extensions) with the community maintaining the application.*

Keywords: *FOSS, free and open source software, open source software, OSS*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the potential benefits, risks, and limits of applying open source software (OSS) and associated support processes to the construction of large systems and to systems of systems. To ensure compliance with federal regulations requiring selection and use of applicable commercial software over new development, they should understand how and where OSS capabilities apply to systems integration, end-user support, and configurability. They should be aware of how OSS compares to other forms of commercial software in terms of acquisition costs, long-term support, scalability, adaptability, security, and resilience in the face of changing requirements. SEs should be aware in particular of the security properties of OSS at the engineering and process levels, and how those properties compare with other types of commercial and government software. They should be aware of the certification status of major OSS packages, and how certification works in the distributed ownership model of OSS. They should understand how to interact successfully and productively with OSS support communities, which use a barter-based economy in which payments are made in terms of software fixes and application capability extensions instead of fees.

Background

Few topics in the software engineering domain of systems engineering, and in engineering information-intensive enterprises, are more likely to engender strong reactions than open source software. Such reactions stem mainly from the community-based ownership model of OSS, in which anyone who agrees to follow an associated OSS license receives the same ownership rights as any other user. This dispersal of engineering change authority violates one of the oldest and most deeply held assumptions of software engineering: High-quality, high-reliability, trustworthy software is possible only if that software has been developed using a well-controlled, authority-centered, top-down development process. OSS not only discards the need for a centralized development authority, but turns the concept upside down by placing control of the development process in the hands of loose collaborations of coders. Since coders are often viewed in software engineering as the participants least likely to understand needs and most likely to violate rules intended to ensure system integrity, quality, maintainability, and security, it is not surprising that a process that relegates change control over to coders would tend to be viewed with distrust.

However, this view is changing for three reasons. The first is the growing realization that just as planned market economies cannot compete with free-market economies at encouraging innovation and efficient use of resources, tightly centralized management of very large software development efforts are more likely to fail than approaches that encourage local innovation and adaptation. OSS encourages such local innovation, and moreover makes the human-readable results of local innovation readily available for any desired level of inspection and analysis.

The second reason is the growing recognition that developing high-quality software unavoidably requires the use of experienced coders who have a strongly mathematical, prove-it-as-you-code-it (“correct by construction”) approach to code development. Just as maintaining the correctness of a complex mathematical proof requires the use of mathematicians who understand those proofs fully, maintaining the correct-by-construction quality features of good software requires the use of experienced coders who understand fully the internal correctness features and properties of that software. OSS development relies on a wiki-like process that encourages continued participation by coders who have the theorem-proving skills needed to maintain and improve the internal correctness of well-designed software. In contrast, non-wiki approaches such as waterfall development actively seek to move code support as quickly as possible to personnel who may be skilled in testing, but who are not usually given an opportunity to learn the structure of the software well enough to maintain its internal correctness properly.

The final reason is pragmatic. OSS use is widespread in both private and government systems, and has been for many years [1]. The communication software (TCP/IP) that first made the Internet possible was OSS, as were many of the early server systems that provided useful data. Microsoft is one of many examples of commercial companies that make extensive use of open source software to build and expand their product line. Internet Explorer is an example of a notable Microsoft utility that is based heavily on OSS. Essentially all modern Apple products, from Macs to iPods and iPhones, are built on OSS with a thin layer of customized software on top. Google is another industry leader that uses OSS heavily both internally and in its commercial products. Apple and Google are also both good examples of how effective use of OSS can enable more and faster innovation by keeping costly designers focused not on maintaining old code, but on developing entirely new capabilities. Finally, nearly every network appliance and custom hardware box sold in the open market today is built mostly or entirely using OSS. OSS is extremely popular with appliance vendors due to its low cost, easy scalability, flexible adaptation to new environments, broad range of available functions, and field-proven reliability.

Government Interest and Use

On October 16, 2009, the U.S. Department of Defense (DoD) issued an updated policy on the use of open source software (OSS) [2]. The policy emphasizes and explains the legal status of OSS as a form of commercial software, which means that it falls under U.S. law (10 USC 2377), Preference for acquisition of commercial items [3]. Not including assessments of OSS options when examining commercial options for reducing costs and improving quality in DoD systems can inadvertently violate this law. A good example of the seriousness of the commitment of the executive branch to assessing, selecting, and using commercial OSS is the White House website http://whitehouse.gov/, which is based in part on the OSS blogging tool [4, 5, 6].

Best Practices and Lessons Learned

Read and understand the U.S. DoD Web page on free and open source software (FOSS) [7]. The U.S. Department of Defense spent years creating three documents analyzing and elaborating the role of OSS in DoD systems. The site addresses DoD policy toward open source, frequently asked questions about the federal role and legal status of open source, and a survey on the widespread prevalence and importance of OSS to the DoD as early as 2003. The Web page is written generically and applies with very little change to other federal departments and agencies. MITRE systems software engineers working with the DoD should in particular make sure they have looked at the October 16, 2009, DoD policy statement at the site.

The larger the community supporting OSS, the greater reduction in long-term support costs. This rule of thumb is at the heart of how OSS can provide significant cost and capability benefits when building large systems. Notably, it has nothing to do with the ability to modify code per se, and in fact can easily be seriously undermined by a premature project interested in modifying OSS source code. Because OSS support works like a consortium, its cost benefits to individual members are highest when the consortium size is as large as possible. These cost benefits increase even further if the OSS support community is large enough to include world-class experts in specific OSS features, since such members often can resolve difficult problems in a tiny fraction of the time that would be required by more generalized support.

Avoid proliferating OSS licenses. There are already far too many OSS licenses. However tempting it may be for an organization to create its own unique OSS license, each license simply further confuses the developers, lawyers, and project managers who must deal with it, and also tends to subdivide the pool of developers available to support such new licenses. Four major license types are typically sufficient:

- **GNU General Public License (GPL):** This popular license requires that any new source code made using GPL source code must itself be licensed as GPL; that is, it must be donated back to the OSS community that created the first source code. Although this feature makes GPL controversial, it also makes it very good at stabilizing the deep infrastructure of a system or network by removing any profit incentive to change it arbitrarily. The Linux kernel was created in part using a GPL license, and demonstrates another feature of GPL: Standard interface to GPL components can be used without any need for the software that uses it to be GPL.
- **GNU Lesser General Public License (LGPL):** This is a variant of the GPL that allows GPL components to be embedded as "library components" in non-GPL code. It is popular with small companies that like the GPL model but do not want to keep businesses from using or buying their software components.

  Note: GNU (GNU's Not UNIX) is a UNIX-like operating system developed by the free software movement starting in 1984. In 1992, the almost-complete GNU system was combined with the Linux kernel, producing the GNU/Linux system. The GNU project developed many of the core programs in GNU but also included available free software such as the X Window System and TeX.
- **Berkeley Software Distribution (BSD)/ Apache:** These forgiving licenses allow companies to "capture" copies of source code and treat those copies and any changes they make to them as proprietary. Apple has made use of this feature of BSD license in creating its current Mac personal computer, iPod, and iPhone product

lines. Due to the high cost of individually maintaining large sets of source code, the majority of participants on BSD/Apache licenses continue to support their OSS products under a community model. For systems engineers, BSD and Apache licenses should be viewed as tools for ensuring that small businesses participating in a large system-of-systems effort will have a strong cost incentive to adapt OSS features provided under a BSD or Apache license. For example, the selection of a BSD-like licensing model for the initial release of the Internet communications software (TCP/IP) was instrumental in getting dozens of small and large businesses with unique networks to accept the code and initiate the first working Internet.

- **No license (government code):** This is the legally required status of code developed by government employees. While approximating a BSD or Apache license by allowing anyone to use it, it can cause considerable confusion if a person or company chooses to copyright the entire work "as is" without acknowledging its government origins.

Don't assume that the lawyers involved will understand OSS licenses. Lawyers who are not deeply familiar with software, and more specifically, how it is converted from readable source code into executable machine code, will have a very difficult time even reading the GPL license and LGPL licenses, let alone understanding them. BSD and Apache licenses avoid details of software structure, and are far easier for lawyers to understand. Often, BSD and Apache are favored by lawyers for that reason alone: They understand them. This unfortunate state of affairs is slowly improving, but in the case of GPL and LGPL, programmers still often understand the meanings and implications of these licenses far better than the lawyers who are responsible for assessing their implications. Systems engineers should be aware of this possible disconnect, and if possible, point lawyers toward relevant documents such as the Frequently Asked Questions (FAQ) [3] on the DoD FOSS website [7].

Use OSS to stabilize shared infrastructure. Infrastructure here means the software components of a large system or system-of-systems that establish basic functions such as networking and data sharing. As demonstrated by the history of the most successful of all system-of-system infrastructure projects, the Internet, using OSS to encourage sharing basic capabilities can be a powerful tool for promoting the emergence of more complex and often unanticipated new capabilities on top of that infrastructure. OSS can also help stabilize large systems by removing the profit incentive for companies to change features arbitrarily or lock customers into unique feature sets. Finally, since infrastructure is often the code that is least innovative, using OSS frees up intellectual resources for more innovative new design work.

Use OSS to help focus costly resources on innovation. The end result of factoring out "solved" problems from large systems and moving them into OSS is shown in the pyramid-like structure in Figure 1. The main concept in this figure is that by factoring out capabilities that are stable, changing relatively slowly, and well-supported by

OSS communities, an organization can pull critically needed designers and coders from support roles. They can move them into more innovative positions focused on the most critical needs of the organization, typically making use of many of the prototyping and exploratory features of OSS (see next two paragraphs).

Encourage use of OSS liaison positions. An OSS liaison is a technically proficient programmer who has been tasked to track, participate in, and make use of a related suite of OSS applications. An experienced OSS liaison both helps make sure that the needs of an organization are understood and sustained by its support community, and provides quickly available internal advice on whether and how a combination of OSS capabilities might meet or support an identified system-level need. OSS liaison positions are non-standard in terms of standard software engineering job profiles, but provide one of the most effective approaches to ensuring that a broad need does not end up being translated inappropriately into a long-term software development project that will at best only replicate features already available through OSS.

Understand the advantages of OSS for exploratory prototyping. One of the most powerful features of OSS is its ability to support experimental prototyping, including research development of new features. Because OSS is developed and supported by distributed groups consisting mostly of individual coders, new features tend to get generalized quickly to make them usable by the entire group of OSS supporters. When this effect is multiplied through multiple levels of code and across many types of systems, the result tends to be an overall set of capabilities that is unusually easy to combine in new ways and adapt to new situations. Since the source code is available, it is also far easier for developers to understand how critical features work and the concepts behind them. All of these features make OSS

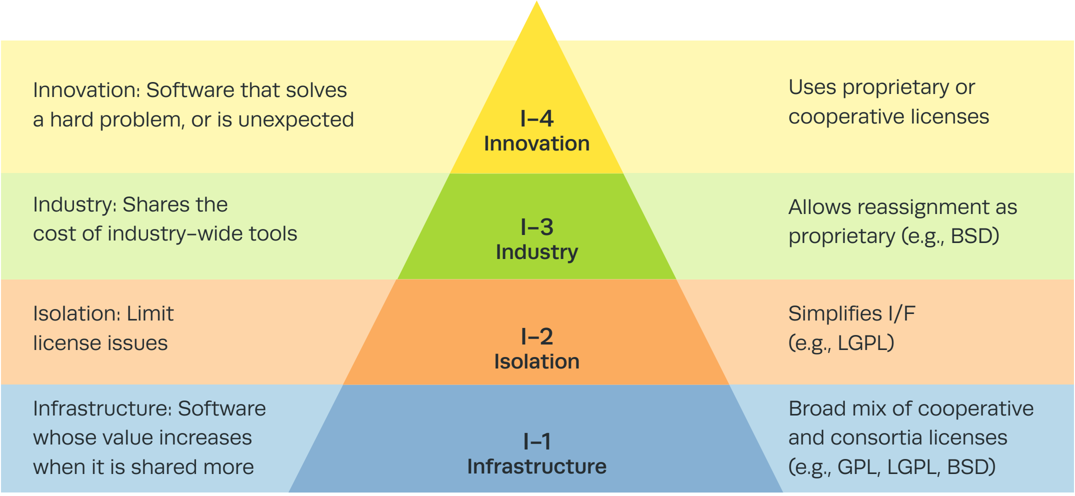

Innovation: Software that solves a hard problem, or is unexpected
I-4
Innovation
Uses proprietary or cooperative licenses
Industry: Shares the cost of industry-wide tools
I-3
Industry
Allows reassignment as proprietary (e.g., BSD)
Isolation: Limit license issues
I-2
Isolation
Simplifies I/F (e.g., LGPL)
Infrastructure: Software whose value increases when it is shared more
I-1
Infrastructure
Broad mix of cooperative and consortia licenses (e.g., GPL, LGPL, BSD)

Figure 1. The I-4 Architecture Pyramid

exceptionally well suited for exploratory prototyping and research into how to build entirely new capabilities. Apple iPhones are a good example of how the highly composable interface capabilities of OSS made it easier for Apple to prototype and develop new approaches to interacting with a mobile phone unit.

Understand the advantages of OSS for systems and systems-of-systems integration. For many of the same reasons that OSS makes a powerful exploratory prototyping tool, it also provides a powerful approach to handling the integration problems typical of large and complex systems engineering problems. OSS includes packages and languages designed specifically to help translate between diverse and often unexpected types of data and communication protocols. One of the most important advantages of such translation-supporting OSS tools for large, complex systems is that they can be used to help simulate the inputs and interactions expected by older and out-of-date components of such systems. Since changing such older components is often both very costly and highly risky, the ability to build such "soft" interfaces to older systems while maintaining current protocols and standards in the rest of a system-of-systems can be very valuable for minimizing overall levels of risk to the project.

Treat OSS licenses like any other proprietary licenses. It would be very unusual for a large federal development project to consider seriously violating the license agreements it has made with large proprietary software companies such as Oracle, IBM, or Microsoft. Yet, ironically, and in part due to widespread misconceptions that OSS means "free" software with no strings attached whatsoever, it is surprisingly common for developers to violate OSS licenses, such as by stripping OSS licenses from source code. This is not just an extremely bad idea from a development quality and long-term support perspective, it is also illegal, unethical, and can result in legal action from watchdog groups such as the Free Software Foundation (FSF) [8]. More important, it undermines the consortium-style share-and-contribute model that is where the real cost reduction potential of OSS resides. Systems engineers should do what they can to ensure that on any given project, OSS licenses will be treated with the same degree of respect they would give to any other commercial license.

When building large systems, try to minimize the need for new software. Historically, software projects have used lines-of-code written as a way of gauging schedule progress, which has resulted in a tendency to think that more code is a good thing. However, because every new line of code entails a long-term cost and reliability burden that may endure for decades, the real goal should be the opposite: Systems engineers should always look for ways to reduce the need for new, unique software to an absolute minimum. Code usually will be needed, but it should be relatively small in size, powerful in capabilities, and uniquely specific to the system being created. Common features such as ordinary file access or standard network communications never fall within this category, and should always be handled by instead acquiring stable, well-standardized OSS or proprietary software.

Strongly discourage any view of OSS as "free source code" for speeding up internal development. Long-term code support costs always dwarf initial software costs for any type of software. For this reason alone, viewing OSS as "free" source code to speed up short-term development goals is a short-sighted and frankly dangerous perspective. A good analogy is this: If your organization was offered all of the source code for Microsoft Windows without charge, but with the stipulation that you would have to fix all bugs and make all enhancements yourself from then on, would you accept the offer? Large OSS packages are at least as complex as Windows, so casually adopting such source code into small internal projects creates the support-cost equivalent of a very large and fast-ticking time bomb. A useful three-step rule of thumb is this: Try executables first, community support second, and new code last:

- **Always try using the executable version of the OSS first:** OSS tends to be more configurable than alternatives, so the first step is to consult with experts to see if an OSS application can be configured to meet a need simply by downloading the "standard release" binary form of the OSS. (For security reasons, you may still want to download and compile the source code locally.)
- **Next, try submitting non-sensitive changes to the supporting community:** If some feature of an OSS application absolutely must be changed or extended to the source code level, the next step is to try to express the changes needed in a way that can be submitted directly to the community that is supporting the application. This approach not only reduces the need for long-term support of the source code but can also help build a stronger relationship with the supporting community.
- **As a last resort only, develop your own modules:** In rare cases where code changes absolutely cannot be made public, look for ways to develop independent modules. If possible, avoid any inclusion of OSS source in such modules. Every line of OSS source code in a new module must be checked and updated whenever the original OSS is updated, and it can also needlessly complicate the legal status of the module.

Avoid overly casual release of government code as OSS. Government projects responsible for nonsensitive government off-the-shelf (GOTS) products often find the community support features of OSS highly attractive and wonder whether they can simply release their GOTS products under OSS licenses to take advantage of the lower costs, faster bug fixes, and improved long-term support seen in some OSS projects. The answer to this question is easy: No. The valuable properties of OSS support emerge from having a community of people already interested in the product, and the valuable modularity and flexibility of OSS emerges from it having been developed over a period of years by such a community. Simply making GOTS products OSS by changing their license and posting them on a website exposes all of their flaws to any interested hackers, without necessarily attracting the interest of supporters who will in any case almost certainly be mostly baffled by the unfamiliar source code. A better approach to replacing GOTS applications is to look for configurations of existing OSS tools that could be used to approximate the

GOTS features. Then try to start building a new community around that configuration to build it up into a full-featured analog or extension of the original GOTS tool.

Encourage a realistic understanding of security in all commercial software. If software is sold or released in the form of binary code, its security situation in the modern world is no different from software that has been released in the form of human-readable source code. The reason is that modern hacking tools work directly against the binary forms of software to attempt to crack it, making the binary form in some ways preferable over the human-readable form that would be hugely slower to analyze. Thus the commonly expressed fear that OSS cannot be made secure because "the source code is available" is just nonsense.

Conversely the argument that OSS is always more secure because "thousands of eyes" are looking at it is also faulty for a simple reason: Just because source code is posted on a website doesn't mean anyone is looking at it at all. Proprietary software may also be touted as more secure because it has been "certified" in one way or another. Unfortunately because no software certification processes in existence has ever been demonstrated in a scientifically assessed field study to produce software that is measurably more secure or reliable than uncertified software, it is not clear that such certifications mean anything beyond that the companies involved were able to afford the high cost of such certifications. Certifications are applied inconsistently, with federal desktops typically running systems and applications that have never been certified, or which were certified so long ago that the certifications no longer apply. OSS is sometimes assumed to be non-secure because "anyone can insert a change" into the code. Although it is true that anyone can make a change to their own copy of OSS source code, in actuality, large, active OSS projects such as Linux have closely guarded and highly automated code control processes that only allow code to enter into the main build process after it has been scrutinized literally by some of the world's top experts on operating system kernels—a level of verification that would put most proprietary code control and review processes to shame.

Conversely, many proprietary processes that keep source code secluded are riddled at multiple levels with opportunities for people to insert changes that could gravely damage the security of such systems. The bottom line is this: Security is a process that is best assessed based on the actual details of each case; whether it was developed using proprietary or OSS community methods changes the issues that must be examined. Proprietary methods have the advantage of bringing more money to bear, while community methods are more visible to more users. When active and global in scope, OSS can also bring to bear experts who are far more likely to spot infiltration attempts.

Software certifications: Look for them, support getting them, but never rely on them. As noted earlier, there is no scientific evidence that software certifications make any difference in the field-level reliability or security of software. They are nonetheless required in many cases. For OSS, companies such as IBM have helped provide certifications. Systems engineers therefore should look for certifications of relevant OSS in case they are available, and see how they compare to proprietary equivalents. It is also possible for interested projects to help OSS groups get certifications, such as through the small proprietary companies that often handle the business side of the use of a particular OSS component. Finally, regardless of whether a commercial software component is OSS or not and certified or not, the overall security of a networked software system should never be assumed to be proven; multiple layers of security are an absolute necessity.

References and Resources

1. The MITRE Corporation, 2003, Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense.
2. Department of Defense, October 16, 2009, Clarifying Guidance Regarding Open Source Software (OSS).
3. DoD NII-CIO, DoD Open Source Software Frequently Asked Questions, accessed February 12, 2010.
4. O'Reilly, T., October 25, 2009, "Thoughts on the Whitehouse.gov Switch to Drupal."
5. The White House, October 24, 2009. (The White House website switched to OSS.)
6. Drupal.org (The White House website switched to OSS), accessed February 12, 2010.
7. DoD NII-CIO, Free Open Source Software (FOSS), accessed February 12, 2010.
8. Free Software Foundation, accessed February 12, 2010.

Additional References and Resources

"Berkeley Software Distribution," Wikipedia, accessed February 12, 2010.

Clarke, G., October 27, 2009, "US DoD snuffs open-source 'misconceptions'," *The Register.*

Coopersmith, A., November 24, 2009, "Sun relicensing to current X.org license," Sun.com.

FORGE.MIL, accessed February 12, 2010.

Hart, K., October 27, 2009, "Defense Department wants more open source software," *The Hill*.

Linux Distributions—Facts and Figures, DistroWatch.com, accessed February 12, 2010.

Military Open Source Software, accessed February 12, 2010.

Open Source FAQ, Microsoft, accessed January 21, 2011.

Open Source Initiative, accessed February 12, 2010.

"Open-source license," Wikipedia, accessed February 12, 2010.

Ryan, J., November 25, 2009, "Sun Leaves License Behind," *Linux Journal*.

SourceForge, About SourceForge, SourceForge Directory, accessed February 12, 2010.

ENGINEERING INFORMATION-INTENSIVE ENTERPRISES

Privacy Systems Engineering

Definition: *Privacy is individuals' claim to determine when, how, and to what extent their information is communicated [1]. Privacy concerns the collection, use, maintenance, and disclosure of personally identifiable information (PII)—any information any agency has about an individual, including information that can be used to distinguish or trace an individual's identity (name, SSN, date and place of birth, mother's maiden name, biometric records) or that is linkable to an individual (medical, educational, financial, and employment records) [2].*

Keywords: *E-Government Act, Fair Information Practices (FIPs), personally identifiable information (PII), privacy, Privacy Act, privacy impact assessments (PIA) record, system of record*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the basic concept of privacy and be able identify PII and the situations in which privacy issues may arise. They should understand the legal requirements that apply to federal agencies' collection, use, maintenance, and disclosure of PII, and how these requirements relate to the systems engineering life cycle (SELC). Further, systems engineers are expected to develop, implement, and maintain technical controls to be included in information technology (IT) systems, which help ensure that privacy requirements are met.

Background

Privacy is based on the implementation of Fair Information Practices (FIPs), which were initially developed in 1973 by a federal advisory committee, commissioned because of concern over the harmful consequences that computerized data systems could have on the privacy of personal information. A revised version of the FIPs, developed by the Organization for Economic Cooperation and Development in 1980, has been widely adopted and forms the basis for many privacy laws worldwide (see Table 1) [3].

Table 1. The Fair Information Practices

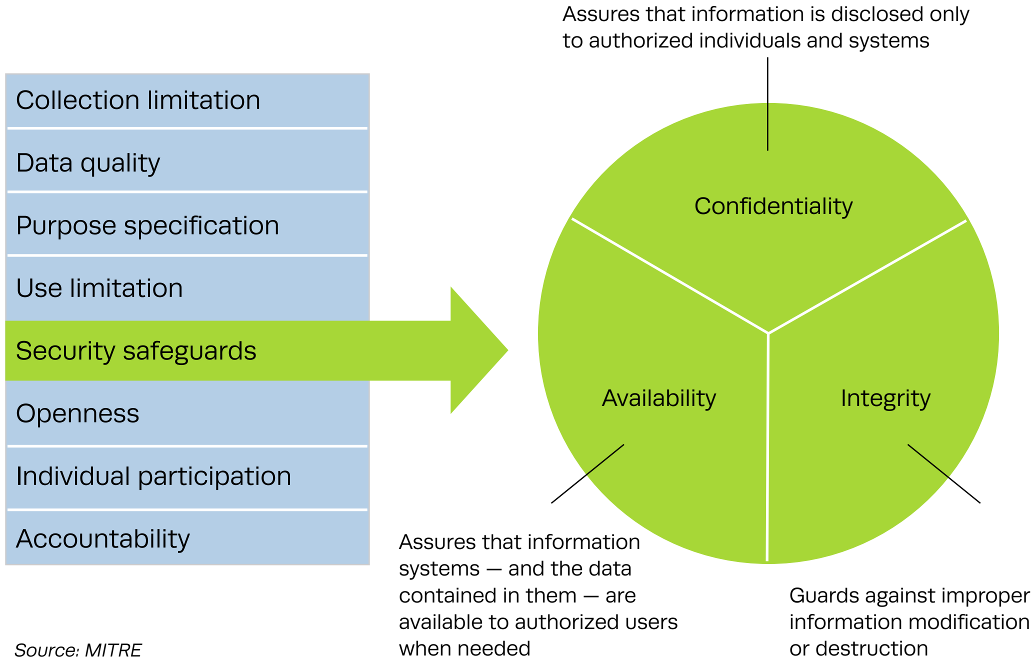
Principle	Description
Collection limitation	The collection of personal information should be limited, should be obtained by lawful and fair means, and, where appropriate, with the knowledge or consent of the individual.
Data quality	Personal information should be relevant to the purpose for which it is collected, and should be accurate, complete, and current as needed for that purpose.
Purpose specification	The purposes for the collection of personal information should be disclosed before collection and upon any change to that purpose, and its use should be limited to those purposes and compatible purposes.
Use limitation	Personal information should not be disclosed or otherwise used for other than a specified purpose without consent of the individual or legal authority.
Security safeguards	Personal information should be protected with reasonable security safeguards against risks such as loss or unauthorized access, destruction, use, modification, or disclosure.
Openness	The public should be informed about privacy policies and practices, and individuals should have ready means of learning about the use of personal information.
Individual participation	Individuals should have the following rights: to know about the collection of personal information, to access that information, to request correction, and to challenge the denial of those rights.
Accountability	Individuals controlling the collection or use of personal information should be accountable for taking steps to ensure the implementation of these principles.

Source: Organization for Economic Cooperation and Development

The centerpiece of the federal government's legal framework for privacy protection, the Privacy Act of 1974, is based on the FIPs and provides safeguards for information maintained by federal agencies. Specifically, the act places limitations on agencies' collection, disclosure, and use of personal information maintained in systems of records. The act describes a "record" as any item, collection, or grouping of information about an individual that is maintained by an agency and contains his or her name or another personal identifier. It also defines "system of records" as a group of records under the control of any agency from which information is retrieved by the name of the individual or by an individual identifier. The Privacy Act requires that when agencies establish or make changes to a system of records, they must notify the public through a notice in the Federal Register. Agencies are allowed to claim exemptions from some of the provisions of the Privacy Act if the records are used for certain purposes, such as law enforcement. However, no system of records can be exempted from all of the Privacy Act's provisions. For example, agencies must publish the required notice in the Federal Register for all systems of record, even those that involve classified information. This ensures that the federal government does not maintain secret systems of records—a major goal of the act [4].

More recently, in 2002, Congress enacted the E-Government Act to, among other things, enhance protection for personal information in government information systems or information collections by requiring that agencies conduct privacy impact assessments (PIAs). A PIA is an analysis of how personal information is collected, stored, shared, and managed in a federal system; it is used to identify privacy risks and mitigating controls to address those risks. Agencies must conduct PIAs (1) before developing or procuring information technology that collects, maintains, or disseminates information that is in identifiable form, or (2) before initiating any new data collections of information in an identifiable form that will be collected, maintained, or disseminated using information technology if the same questions are asked of 10 or more people [5]. Individual PIAs may vary significantly depending on the specifics of department/agency guidance and the scope and sensitivity of PII collected, used, and disseminated.

Note that privacy is not synonymous with security. (See Figure 1.) Whereas privacy focuses on the individual's ability to control the collection, use, and dissemination of their PII, security provides the mechanisms to ensure confidentiality and integrity of information, and the availability of information technology systems. The concepts of privacy and security, however, do intersect. Specifically, certain IT controls established to ensure confidentiality and integrity from a security perspective also support privacy objectives. For example, access controls ensure that only authorized individuals can read, alter, or delete data. Such controls help achieve confidentiality and integrity from a security standpoint. In addition, when a system processes or stores PII, these IT controls ensure that users can access only


Collection limitation
Data quality
Purpose specification
Use limitation
Security safeguards
Openness
Individual participation
Accountability
Assures that information is disclosed only to authorized individuals and systems
Confidentiality
Availability
Integrity
Assures that information systems — and the data contained in them — are available to authorized users when needed
Guards against improper information modification or destruction

Source: MITRE

Figure 1. Privacy vs. Information Security

the specific PII needed to perform their jobs; this helps ensure that use of PII is limited to authorized purposes (purpose specification) and protected from unauthorized access, destruction, and disclosure (security safeguards). Although establishing good security practices helps protect privacy, these practices are not, in and of themselves, sufficient to fully address the FIPs.

Best Practices and Lessons Learned

Privacy must be "built into" the systems engineering life cycle. Consideration of privacy, including requirements to conduct PIAs, should be built into the agency systems engineering life cycle. This helps ensure that privacy requirements are considered early in the development of IT systems and that the technology is leveraged to provide privacy protections. At a minimum, the SELC should include the requirement to conduct a PIA as early in the development process as practicable. In addition, the SELC should contain procedures that require the completion of the PIA before the system is authorized to operate. Considering privacy requirements early in the development process avoids difficult and expensive retrofitting to address them later.

All appropriate stakeholders must be involved in assessing privacy risks. In conducting the PIA, it is critical that the privacy office, the systems developer, and the business process owner all be involved. Having the appropriate stakeholders involved ensures that all privacy risks are identified and that alternative mitigating controls are considered.

Technology can be leveraged to protect privacy. Although many of the privacy risks identified through the PIA will be mitigated by establishing administrative controls—such as providing additional public notice, establishing policies and procedures to allow individuals access to the information held about them, or providing privacy training to system users—certain risks can be mitigated by technical system controls. Table 2 provides examples of how a system should be designed to protect privacy.

Table 2. How System Engineers Can Implement FIPs

Principle	Guidance for Systems Engineers
Collection limitation	Design the system to use only the minimum amount of PII necessary to accomplish the system's purpose. The key question to ask for each field of PII is: Can the purpose of the system be served without this particular field of PII?
Data quality	Develop the system to meet the data quality standards established by the agency.
Purpose specification	Develop systems that interact directly with the public such that the purpose for the collection of PII is made available.
Use limitation	Develop the system such that each field of PII is used only in ways that are required to accomplish the project's purpose. Each process associated with each field of PII should be reviewed to determine whether that use directly fulfills the project's purpose. If not, the function should not be developed.
Security safeguards	Implement information security measures for each field of PII to prevent loss, unauthorized access, or unintended use of the PII. Use encryption, strong authentication procedures, and other security controls to make information unusable by unauthorized individuals. Note: OMB guidance directs that only cryptographic modules certified by the National Institutes for Standards and Technology (NIST) are to be used. See NIST's website at http://csrc.nist.gov/cryptval/ for a discussion of the certified encryption products [7].

Openness	Design the system to provide both a security and privacy statement at every entry point. Develop mechanisms to provide notice to the individual at the same time and through the same method that the PII is collected; for example, if PII is collected online, notice should also be provided online at the point of collection.
Individual participation	Design the system to allow identification of all PII associated with an individual to allow correction of all PII, including propagating the corrected information to third parties with whom the information was shared.
Accountability	Accountability can be encouraged, in part, by the use of audit logs that are capable of supporting a comprehensive audit of collection and use of all fields of PII to ensure that actual collection and use is consistent with the notice provided. Audit logs should not contain the actual content of fields of PII, to limit unnecessary disclosure of this information. The audit log should contain sufficient detail to identify (1) the source of each field of PII, (2) when each field of PII was accessed, (3) the uses of each field of PII, and when and by whom this information was used, (4) when each piece of PII was last updated and why, and (5) any suspicious transactions related to any field of PII and, if these occurred, the nature of the suspicion and the specific users involved. If the use of Social Security numbers (SSNs) is authorized, systems engineers should create mechanisms to log access to SSNs and implement periodic reviews of the audit logs for compliance with the authorization. The audit logs should also record sufficient information to support an audit of whether each field of PII was shared pursuant to a determination that the recipients needed the field of PII to successfully perform their duties, possessed the requisite security clearance, and provided assurance of appropriate safeguarding and protection of the PII.

Source: MITRE and Department of Homeland Security [6]

Note that this list is not intended to be exhaustive. The specific risks and mitigating controls for an IT system or information collection should be identified by conducting a PIA.

Finally, systems engineers should consider the use of enterprise privacy-enhancing technologies (ePETs). ePETs are enterprise-oriented, data stewardship tools that help organizations achieve their business goals while appropriately managing PII throughout the information life cycle. These technologies may or may not be privacy-specific. ePETs include tools that can desensitize static data in databases by applying a variety of transformations, including masking and obfuscation. Desensitized data can then be used for testing and other purposes without unnecessarily disclosing individuals' PIIs. Another example is enterprise digital rights management, which can be used to impose limitations on the usage of digital content and devices, and can also be used to enforce limitations on the use of PII [8].

References and Resources

1. Westin, A., 1967, *Privacy and Freedom*, NY: Athenaeum.
2. Office of Management and Budget (OMB), July 12, 2006, Reporting Incidents Involving Personally Identifiable Information, M-06-19.
3. Office of Economic Cooperation and Development, September 23, 1980, OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.
4. The Privacy Act of 1974, as amended, 5 U.S.C. § 552a.
5. The e-Government Act of 2002, Public Law 107-347.
6. Privacy Office of the Department of Homeland Security, August 16, 2007, Privacy Technology Implementation Guide.
7. OMB, May 7, 2007, Safeguarding Against and Responding to the Breach of Personally Identifiable Information, M-07-16.
8. Shapiro, S., February 27, 2009, A Gentle Introduction to Privacy Enhancing Technologies, PowerPoint Presentation.

Additional References and Resources

Cannon, J. C., 2004, Privacy: *What Developers and IT Professionals Should Know,* Addison-Wesley.

McEwen, J., and S. Shapiro (eds.). 2009, *U.S. Government Privacy: Essential Policies and Practices for Privacy Professionals.* York, ME: International Association of Privacy Professionals.

MITRE has also developed several methodologies to support privacy engineering. For example, Privacy-Based Systems Analysis (PBSA) supports systematic strategic privacy analysis and planning at the program or mission level. It enables organizations to rigorously consider the alignment between privacy objectives and business and technical architectures. Privacy Risk Maps provide a view of enterprise privacy risk by documenting where PII resides and how it moves within the organization and across organizational boundaries. MITRE has also developed the Privacy RIsk Management Engine (PRIME), a Web-based PIA tool to support more effective privacy risk analysis and mitigation.

Systems Engineering for Mission Assurance

Definition: *Mission assurance means operators achieve the mission, continue critical processes, and protect people and assets under internal/external attack (physical and cyber), unforeseen environmental or operational changes, or system malfunction. Across the acquisition life cycle, systems engineering for mission assurance enables awareness of changing adversarial strategies and environmental and system conditions, options to achieve a mission under different circumstances, tools to assess and balance advantages/risks of options, and transition to an option while continuing the mission.*

Keywords: *assurance, attack, cyber, dependability, information, mission, operational, quality, resilience, risk, success, supply, threat*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be conversant in mission operations, advanced persistent threats, unforeseen environmental changes, and system malfunctions that can cause missions to fail. They are expected to be familiar with the basic principles for building and operating systems that can sufficiently fight or operate through these obstacles. They should be knowledgeable in the effects that the mission/operators are attempting to achieve and the various options and alternatives that systems or combinations of systems can provide to achieve these effects. MITRE SEs need to understand methods for determining vulnerabilities, countermeasures, and residual risks to mission accomplishment based on available system options and alternatives. They are expected to be able to effectively convey these methods and the results of applying them to system stakeholders and decision makers. MITRE SEs are expected to recommend requirements, strategies, and solutions for mission assurance capabilities, including consideration of technical and operational dimensions across all phases of the system life cycle, from concept development through deployment and operations. They are expected to encourage and facilitate active participation of end users and other stakeholders in developing capabilities that will ensure mission success. They are expected to monitor and evaluate contractor mission assurance technical efforts and recommend changes when warranted. MITRE SEs are also expected to keep abreast of the evolving discipline of systems engineering for mission assurance.

Context

The concept of engineering a system that can withstand purposeful or accidental failure or environmental changes has a long history in the discipline of designing systems for survivability. In the Cold War era, designing for survivability meant nuclear hardening of command centers, creating alternate command centers, engineering electronic countermeasures into communications and sensor systems, building redundant backup components, and engineering fallback modes of operation and switchover capabilities among them. More recently, the notion of engineering for mission assurance has been extended to ensuring the ability to effectively operate at the "tactical edge" in an environment with limited, austere, or intermittent communications, by selecting from a variety of communications options in the event that primary means become inoperable. Designing communications systems for survivability meant redundant communications links and factoring in potential adversary actions such as electronic warfare. Although all these are still needed in the Internet era, engineering systems for mission assurance has been further expanded to include engineering for information assurance and cyber security.

In recent years, cyber threats have become the predominant focus of mission assurance. Some worry that such intense focus on "all things cyber" risks losing sight of other dimensions of mission assurance. Others see a tension or conflict between mission assurance's "get the operational job done" ideal of achieving 100 percent mission success every time and the security-focused aims of information assurance, which could at times constrain aspects of operations in order to protect data and systems. Yet others are concerned that the acquisition community does not have a sufficiently mature mission assurance culture or mindset and that we are not yet sufficiently attuned to mission assurance as an "implicit requirement" that needs to be considered for all systems, whether or not it is explicitly demanded.

When we engineer for mission assurance, what essential attribute are we seeking to "engineer in"? Is it robustness, resilience, dependability, risk management, security, agility, flexibility, or adaptability? Is it one of them, some of them, or all of them? What are the tradeoffs and how are they determined? Who is the decision maker—the operator, the overseer, or the accreditor—and what role should each play in the decision-making process? What does "systems engineering for mission assurance" look like? The reality is that we don't yet have a complete answer. But we do have partial answers, and we are continuously evolving our understanding and practice of it every day. What we do know is that, taken together, the various dimensions of mission assurance pose some of the most difficult challenges in engineering systems today.

The working definition of "systems engineering for mission assurance" in this guide is rooted in the insight that *operational users of military systems are almost always willing to accept some level of risk in accomplishing their missions*. It is in the nature of their profession, but to do that, they need the tools to understand the risks they are accepting and the ability to assess and balance available options and alternatives. This suggests that "systems engineering for mission assurance" is the art of engineering systems with options and alternatives to accomplish a mission under different circumstances and the capability to assess, understand, and balance the associated risks. Options and alternatives will likely take the form of a blend of technical and operational elements, which requires systems engineer to have an intimate understanding of the technical details and limitations of the system, the doctrine and operations of the user, and the environmental conditions and threats that will or may be encountered.

Articles Under This Topic

The articles under this topic are focused on what we know today about systems engineering for mission assurance. It is a rapidly evolving field, so check back often for updates and additional material.

"Cyber Mission Assurance" structures the cyber response discussion around the notion of a system architecture that is resilient in the face of different levels of cyber threat. The article focuses on near-term actions, rooted in actual experience, to begin evolving architectures that reduce their attack surface and are more secure, resilient, understandable, agile, and manageable.

The next three articles step through the elements of the mission assurance engineering (MAE) process. "Crown Jewels Analysis (CJA)" is a methodology that helps identify the cyber assets most critical to mission accomplishment—the "crown jewels" of a crown jewel analysis—and that begins during system development and continues through deployment. "Cyber Threat Susceptibility Assessment" helps understand the threats and associated risks to those assets. "Cyber Risk Remediation Analysis (RRA)" is used to identify and select mitigation measures to prevent or fight through cyber-attacks.

"Secure Code Review" provides an overview of the specialized task of automatically or manually reviewing security-related weaknesses of an application's source code to understand what classes of security issues are present. The goal of a secure code review is to arm the systems engineer and code developer with information to make an application's source code more sound and secure.

"Supply Chain Risk Management" discusses the threats to and vulnerabilities of commercially acquired information and communications technologies that government information and weapon systems use. It discusses how to minimize the risk to systems and their components from sources that are not trusted or identifiable, or that provide inferior materials or parts.

References and Resources

MITRE Digest, May 2010, "Mission Not Impossible," The MITRE Corporation.

Guerro, S. B., and W. F. Toseny, eds., The Aerospace Corporation, 2007, Mission Assurance Guide, TOR-2007(8546)-6018 Rev. A.

Gupta, Rahul, 2006, The Need for Mission Assurance, PRTM.

National Defense Industrial Association (NDIA) System Assurance Committee, 2008, Engineering for System Assurance, Arlington, VA.

Definition: *Mission assurance is "a process to ensure that assigned tasks or duties can be performed in accordance with the intended purpose or plan ... to sustain . . . operations throughout the continuum of operations" [1]. It is executed through a "risk management program that seeks to ensure the availability of networked assets critical to department or agency missions. Risk management activities include the identification, assessment, and security enhancement of assets essential for executing ... national ... strategy" [1]. Cyber mission assurance focuses on threats resulting from our nation's extreme reliance on information technology (IT).*

Keywords: *architecture, cyber, cyber threat, mission assurance, resilience*

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Cyber Mission Assurance

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to help customers acquire robust systems that can successfully execute their mission even when under attack through cyberspace. To do this, MITRE SEs are expected to be conversant in mission operations, the various types of cyber threats to IT systems, and system malfunctions that can cause missions to fail. They are expected to be familiar with best security practices and the basic principles for building and operating systems that can sufficiently fight or operate through these obstacles, including architecture resilience against the upper end of the cyber threat spectrum. Given the complexity, variety, and constant change of cyber threats, MITRE SEs should seek out MITRE cyber security experts to support these activities.

They are expected to recommend requirements, architectures, strategies, and solutions for cyber protection and mission assurance capabilities, including consideration of technical and operational dimensions across all phases of the system life cycle, from concept development through deployment and operations. MITRE SEs are also expected to keep abreast of the evolving discipline of engineering for cyber mission assurance.

Background and Introduction

Today's information technology (IT) environments are increasingly subject to escalating cyber attacks. Cyber threats vary widely in sophistication, intent, and consequences to the targeted systems and networks. The range of attackers extends from users who unintentionally damage systems to hackers, cyber criminals, and full-scale cyber spies and cyber warriors; their intentions span from annoying vandalism to economic threats to taking out the electric grid or defeating armed forces. Similarly, the target of the attacks can vary from a single computer or router to an entire online banking system, business enterprise, or global supply chain. At the same time, missions and businesses fall along a spectrum of criticality—from desirable to necessary, essential, and mission- or safety-critical. Given the broad spectrums of threat, intent, and consequence to mission-critical functions, determining exactly where mission systems lie in this continuum of dimensions is vital to determine the appropriate level of investment and response.

The notion that 100 percent cyber protection can be achieved is not only unrealistic but also results in a false sense of security that puts our missions and businesses at serious risk. Consequently, the inability to achieve full protection must be compensated by ensuring that missions can be accomplished despite cyber attacks.

When engineering systems for cyber mission assurance, the focus needs to build upon engineering defensive capabilities via protection technologies and engineering both offensive and defensive capabilities within a comprehensive framework of risk management. Engineering for cyber mission assurance requires a mindset akin to the air traffic control (ATC) system concept of "graceful degradation." Weather cannot be controlled, so the air traffic controllers plan how to gracefully degrade the flow of air traffic during bad weather, and the ATC systems are engineered to enable the execution of those plans. It also calls for addressing the unexpected and the undetectable cyber attack in ways that make the adversary's exploit harder and more costly, less likely to succeed, and more likely to cause minimal impact on mission operations.

Cyber defenses generally available today help address low-end threats but alone are often ineffective against more capable forms of cyber attacks that may target our most mission-critical systems. It is at the high end of the continuum that resilience of the system architecture will matter most—to enable continuity of mission-critical operations and support rapid reconstitution of existing or minimal essential capabilities or the deployment of alternative means of accomplishing the mission.

Thus, although this article presents ideas along the full spectrum of cyber security, it concentrates on architectural resilience against the upper end of the threat spectrum, where the stakes are high, the mission or business is critical, and the adversary is sophisticated, motivated, and persistent.

Nevertheless, many of the same techniques are valuable at the low to medium levels of threats and consequences because they can significantly reduce the operational impact and cost of cleanup after an attack. Even if the intentions and consequences of a threat are currently not very serious, it must be kept in mind that today's massive data thefts or passive reconnaissance can quickly escalate into data and system modification, surreptitious commandeering of control, or denial of essential services with far more dire mission impact in the future.

Some of the recommendations in this article are clearly in the domain of systems engineers or designers, while others may fall more naturally to program managers, cyber security experts, or operational users and their leadership. Cyber mission assurance recommendations are most effective when used in combinations to achieve an overall security strategy. The recommendations are therefore presented as a whole instead of attempting to parse out those that fall under the particular purview of systems engineering.

Last, the specific recommendations that follow will not be practical for all systems. Some can be done in the short term for certain systems, like those now in design, but would take longer or might never be implemented for legacy systems with a large installed base. Some recommendations address a common concern but in different ways, with the expectation that practitioners will find something that makes cost-effective sense for their particular situation. Following any of these recommendations will decrease risk, but the recommendations are best followed in combinations derived from following a security engineering strategy and practice based on modeling the threats the user is trying to protect against.

Three Levels of Cyber Threat

Low-end threats are often known as *hackers* or *script kiddies,* and their techniques typically involve email phishing and hacking. They often take advantage of widely known, still-unpatched vulnerabilities in today's operating systems, applications, and hardware. The motive can be mischief or the bragging rights that come with success. Yet, the same vulnerabilities used by the low-end threat can be used by any threat, including high-end.

Mid-range threats are often state-sponsored and will use low-end techniques to target well-known vulnerabilities where effective. They may also use zero-day attacks (that take advantage of the delay between when vulnerabilities are discovered and when they are reported and corrected); perform reconnaissance and probing to gain knowledge of infrastructure, controls, and configuration weaknesses; and use social engineering to manipulate online users into revealing personal information and other exploits to exfiltrate and/or manipulate information. Mid-range threats can remotely implant malware (viruses, worms, adware, or spyware that can threaten a network) and back doors, cause denial-of-service attacks, and introduce undetectable software modifications that can hide in a system once penetrated and maintain presence across many types of system changes. The cyber espionage documented in a report to the U.S.-China Economic and Security Review Commission is an example of this type of threat [2].

High-end threats use all of the above techniques and add the ability to circumvent physical security measures; create undetectable hardware and software modifications and insert them via the supply chain; plant or turn insiders in the target organization; and use full-spectrum intelligence to identify targets and vulnerabilities.

Responding to Low-End Threats

Some suggestions and resources for managing low-end threats are provided in the paragraph below. More details are at the references cited.

Responding to low-end threats is tedious and costly, but a necessary part of using IT. Using Security Content Automation Protocol (SCAP)-compliant tools will make dealing with low-end threats a more automated process. Even when faced with a mid-range or high-end threat, it makes sense to first deal with the low-end threat, rather than get distracted and let defenses down. Dealing with the low-end threat involves the application of end-to-end solutions incorporating commonly understood components such as firewalls, anti-virus protection, anti-spyware protection, anti-spam protection, intrusion detection, vulnerability patching tools, and scans for wireless access points. *Note*: The Security Content Automation Protocol (SCAP) is a method for using specific standards to enable automated vulnerability management, measurement, and policy compliance evaluation (e.g., FISMA compliance). The National Vulnerability Database (NVD) is the U.S. government content repository for SCAP

The SANS Institute maintains a list of the "Top 20 Internet Security Problems, Threats, and Risks" and what to do about them [3]. MITRE and SANS also produced a list of the "Top 25 Programming Errors" [4]. The SANS website also hosts the Consensus Audit Guidelines (CAG) Twenty Critical Controls for Effective Cyber Defense [5]. However, although commonly understood, these defenses are often either not employed, incompletely employed, misconfigured, or not maintained. *Note*: The SANS Institute, founded in 1989, provides computer security training, professional certification through Global Information Assurance Certification (GIAC), and a research archive—the SANS Reading Room. It also operates the Internet Storm

Center, an Internet monitoring system staffed by a global community of security practitioners. SANS is an acronym for SysAdmin, Audit, Network, and Security.

Responding to Higher Level Threats

For obvious reasons, most detailed suggestions and resources for managing mid- to high-end threats are sensitive, closely held, and often rapidly evolving to keep pace with ever-changing threats. Recently, cyber mission assurance thought leaders have started structuring the cyber response discussion around the notion of a system architecture that is resilient in the face of different levels of cyber threat. The MITRE report Building Secure, Resilient Architectures for Cyber Mission Assurance [6] surveys the emerging thinking on building secure, resilient architectures for cyber mission assurance. It motivates the need, lays out the goals and objectives of a resilient cyber architecture, defines its key characteristics, and provides an overview of key resilience techniques and mechanisms.

Defining Resilient Architectures

The term *resilience* has many definitions depending on the context and application. For a computing paradigm, the simple definition from the University of Kansas's ResiliNets Project proves most useful: "Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation." Resilience is related to survivability, which builds on the disciplines of security, fault tolerance, safety, reliability, and performance.

Government departments and agencies are increasing their attention to resilience. Though this increased attention clearly indicates an understanding of the importance of resilience, the community is just beginning to understand what it means to turn the concept into practice. Much work is needed to define and validate resilience: techniques and strategies; policies to promote operational and system resilience; risk decision methodologies, analytic processes, and acquisition guidance; and metrics for measuring resilience improvements and evaluating progress. Moreover, funding must be aligned to budget cycles to reflect these needs and build momentum.

Game-changing technologies, techniques, and strategies can make transformational improvements in the resilience of our critical systems. A number of the detailed ideas on building secure, resilient architectures for cyber mission assurance in [6] are future-looking and suggest the art of the possible from which to begin evaluating the viability of promising strategies and techniques for resilience, singly and in combination, to determine which are the most cost-effective to pursue.

Best Practices and Lessons Learned

The following four items are among the most mature practices of engineering for cyber mission assurance. They are rooted in actual experience. The references point to more details.

To begin evolving our architectures to be more secure and resilient, the first step is to reduce their attack surface and make them more understandable, agile, and manageable. Near-term re-architecting actions can begin now by addressing these four principles:

Virtualization: Leverage or introduce virtualization as a foundation to implement techniques for isolation, non-persistence, replication, reconstitution, and scaling. Doing so will support capabilities to constrain attacks and damage propagation, improve availability, and provide agility to create, deploy, and move critical capabilities at will (moving target defense) if the system is under attack. ([6], pp. 8, 10, 11–14)

Non-persistence: Non-persistence techniques can be applied for access to data, applications, and connectivity when continuous access is nonessential. They can be used to reduce the exposure of the data and applications, as well as the opportunity for the adversary to analyze our vulnerabilities or gain a stronghold and maintain a persistent presence. Non-persistence can also provide operational provisioning and management benefits by pushing a new gold image when a user connects each day or at some fixed interval, thus reducing the period of vulnerability. The goal is to set the frequency so that refreshes occur often enough to prevent the spread or intended impact of an attack, but not so often that it makes the system unstable. Refreshing aperiodically to a known good image can provide the additional advantage of hindering an attacker’s ability to predict the window of opportunity in which to launch an attack, thus increasing the risk that the attack will fail or be detected, and reducing the likelihood of gaining a persistent stronghold. Aperiodic refreshing may require additional coordination. ([6], pp. 8–9, 12, 14–15)

Partition/isolate: Segregate components of dubious pedigree from trusted ones to reduce the attack surface, simplify systems and interfaces, and limit the damage and spread of exploits, when they occur. Separation requirements should implement the principle of least privilege and separate critical from non-critical mission functions and data. Partitioning supports the distribution and placement of highly specialized sensors that can improve situational awareness and better detect behavioral anomalies ([6], pp. 3–6, 8–9, 11–13). Examples include:

- Separation at the network level (e.g., Internet from Intranet and demilitarized zone segmentation) and at the application and data levels (e.g., non-sensitive from sensitive) segregates risky traffic and processing from critical traffic, processing, and data. These tactics help reduce complexity by removing extraneous data and transactions and promote more effective intrusion detection, packet capture analytics, and other anomaly detection capabilities because anomalous behavior cannot easily “hide in the noise.”

- Segmentation at the network level should implement controlled interfaces between segments, as needed, both *within* an enterprise and at its *boundaries*. This will help limit local area network contamination and flow-through, and with proper sensor deployment can provide better situational awareness (SA) and more precisely targeted computer network defense (CND).
- Isolating the CND network from critical processing networks can help prevent the adversary from learning our intrusion analysis and forensic capabilities.
- Isolating asynchronous communications, analyzing and correlating request-response traffic, and isolating different protocols support detecting anomalous traffic.
- Implementing white lists will constrain application pathways. A white list or approved list is a list or register of entities that, for one reason or another, are being provided a particular privilege, service, mobility, access, or recognition.
- Using secure browsers, thin clients, and virtualized clients to sandbox risky processing from critical processing. In computer security, a sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third parties, suppliers, and untrusted users. The sandbox typically provides a tightly controlled set of resources for guest programs to run in, such as scratch space on disk and memory. Network access and the ability to inspect the host system or read from input devices are usually disallowed or heavily restricted. In this sense, sandboxes are a specific example of virtualization.

Situational awareness: Beef up detection, analysis, correlation, and forensics tools and processes. Improve integrated SA understanding by improving sensor data collection, analytics for security and mission-critical capabilities' health (i.e., better detect degradations, faults, intrusions, etc.), and visualization techniques. Baseline normal critical processing and user behavior, and focus on anomaly detection within this context. Use forensics to drive evolution of CND and operations. ([6], pp. 4, 6, 16)

References and Resources

1. Department of Defense, July 1, 2010, DoD Directive 3020.40 Change 1 "DoD Policy and Responsibilities for Critical Infrastructure."
2. Krekel, B., October 9, 2009, Capability of the People's Republic of China to Conduct Cyber Warfare and Computer Network Exploitation," prepared for The US-China Economic and Security Review Commission, Northrop Grumman Corporation.
3. SANS, September 2009, Top Cyber Security Risks.

4. MITRE/SANS, February 6, 2010, Top 25 Programming Errors.
5. SANS, November 13, 2009, "Twenty Critical Controls for Effective Cyber Defense: Consensus Audit," 20 Critical Security Controls, Ver. 2.3.
6. Goldman, Harriet G., November 2010, Building Secure, Resilient Architectures for Cyber Mission Assurance, The MITRE Corporation.

Additional References and Resources

Defense Science Board, April 2007, 2006 Summer Study on Information Management for Net-Centric Operations, Vol. 1.

MITRE authors, May 5, 2009, Selected MITRE Cyber-Related Research and Initiatives. The MITRE Corporation.

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Crown Jewels Analysis (CJA)

Definition: *Crown Jewels Analysis (CJA) is a process for identifying those cyber assets that are most critical to the accomplishment of an organization's mission. CJA is also an informal name for Mission-Based Critical Information Technology (IT) Asset Identification. It is a subset of broader analyses that identify all types of mission-critical assets.*

Keywords: *Advanced Cyber Threat (ACT), Advanced Persistent Threat (APT), criticality analysis, cyber, fight through, information assurance, mission assurance, mission critical, resilience*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to help customers acquire robust systems that can successfully execute their mission even when under attack through cyberspace. To do this, MITRE SEs are expected to be conversant in best security practices and the basic principles for analyzing and identifying IT assets that are critical to the accomplishment of an organization's mission. They are expected to keep abreast of new and evolving techniques for identifying mission-critical assets.

Background

In a large and complex enterprise, it is difficult to know how problems with a portion of an IT infrastructure may affect the broader operational mission. CJA provides a methodology to help understand what is most critical—beginning during systems development and continuing through system deployment. CJA is often the first step in a Mission Assurance Engineering (MAE) process (see Figure 1), which provides a rigorous analytical approach to:

- Identify the cyber assets most critical to mission accomplishment—the "crown jewels" of CJA.
- Understand the threats and associated risks to those assets—via a subsequent cyber Threat Susceptibility Assessment (TSA) [1].
- Select mitigation measures to prevent and/or fight through attacks—via Cyber Risk Remediation Analysis (RRA) [2], which identifies recommended mitigation measures.

MAE offers a common, repeatable risk management process that is part of building secure and resilient systems [3]. The underlying premise for performing a CJA is that protection strategies focused entirely on "keeping the adversary out" are usually doomed to fail (recall the Maginot Line). The Advanced Cyber Threat (ACT) has sophisticated capabilities

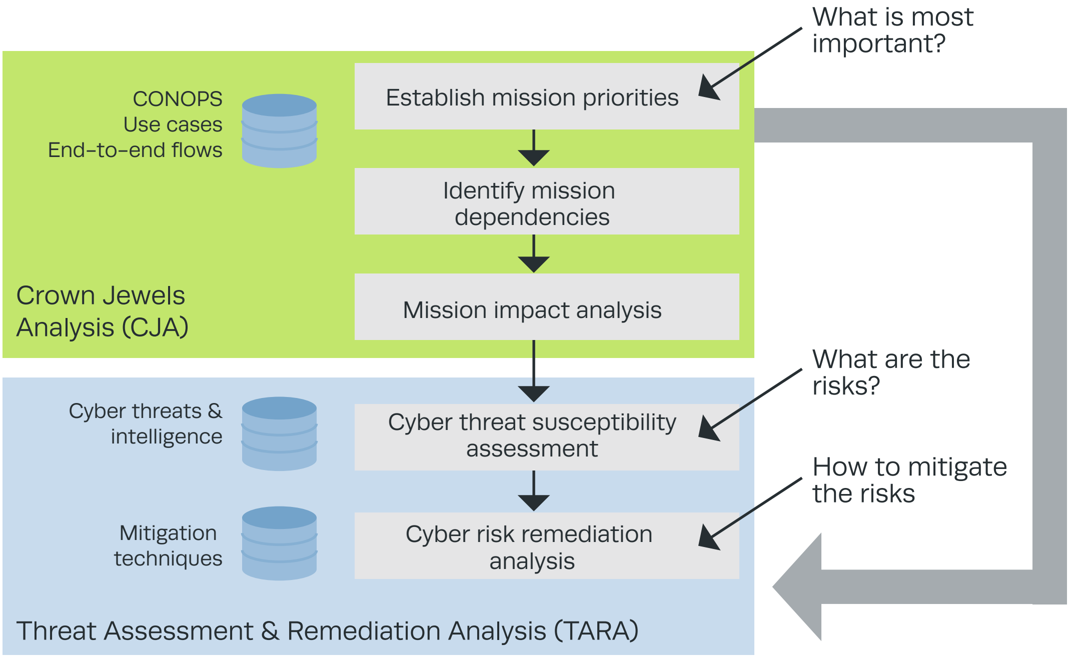

What is most important?
CONOPS
Use cases
End-to-end flows
Establish mission priorities
Identify mission dependencies
Mission impact analysis
Crown Jewels Analysis (CJA)
What are the risks?
Cyber threats & intelligence
Cyber threat susceptibility assessment
How to mitigate the risks
Mitigation techniques
Cyber risk remediation analysis
Threat Assessment & Remediation Analysis (TARA)

Figure 1. The Mission Assurance Engineering (MAE) Process

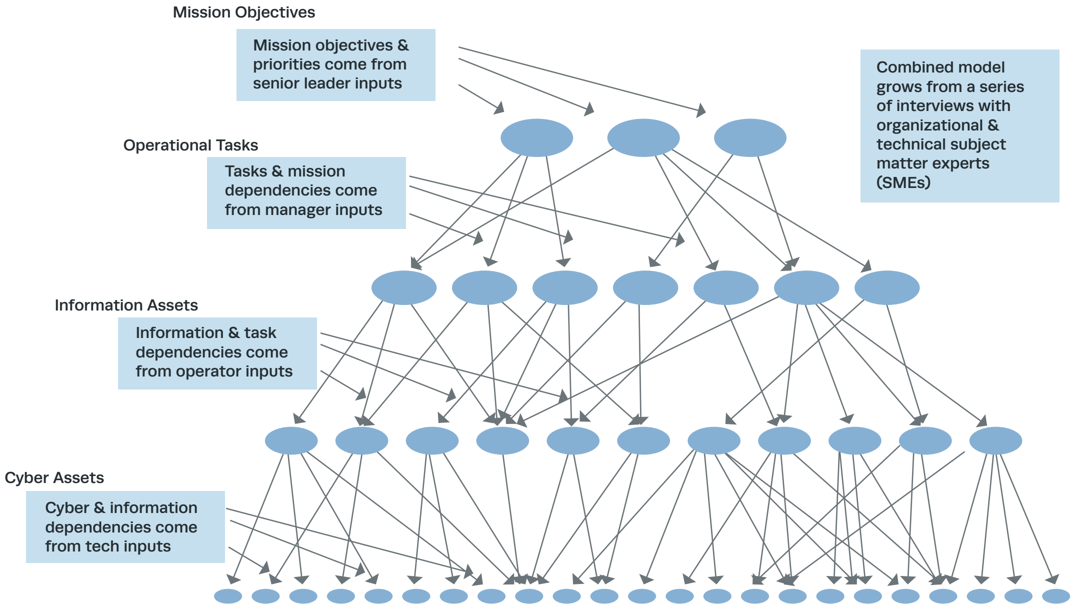

Mission Objectives
Mission objectives & priorities come from senior leader inputs
Operational Tasks
Tasks & mission dependencies come from manager inputs
Information Assets
Information & task dependencies come from operator inputs
Cyber Assets
Cyber & information dependencies come from tech inputs
Combined model grows from a series of interviews with organizational & technical subject matter experts (SMEs)

Figure 2. CJA Model Using Dependency Maps

that adversaries can use to gain and maintain a persistent presence in the hardware and software that make up mission systems—providing the opportunity to "attack" (i.e., deny, degrade, deceive) these assets at a time and place of their choosing. As cyber threats continue to escalate, it is prudent to assume that adversaries will aim to successfully penetrate and then deny and/or degrade our cyber assets, necessitating that we must maintain our ability to "fight through" such attacks. Because it would be prohibitively difficult and costly to design every component of a system to fight through all conceivable attacks, a CJA is used to identify the most important cyber assets to an organization's mission—allowing systems engineers, designers, and operators to focus on ensuring that these critical components can effectively endure an attack.

Organizations (especially operational units) often have limited resources to use in finding their mission-critical cyber assets, and yet they need a reasonably accurate idea of what those are. One technique for performing a CJA makes use of "dependency maps" [4]. This technique stemmed from a need for a convenient but moderately rigorous approach—with more detail and structure than is possible from using a questionnaire. A CJA dependency map uses features adapted from MITRE's Risk-to-Mission Assessment Process (RiskMAP) [4, 5]. As a result, this particular CJA methodology and RiskMAP are often taken as one and the same, but they are not. A more rigorous approach would involve Cyber Mission Impact Assessment (CMIA) [6], which uses a more detailed description of the user's system.

The dependency map method uses facilitated discussions with system subject matter experts (SMEs) to assemble a model, from the top down, as shown in Figure 2 [7].

These dependencies are qualitatively expressed in terms of "If <*child*> fails or is degraded (as defined by the SMEs), the impact on <*parent*> is <*failure, degrade, work-around, nominal*>." Provisions are included to minimize subjectivity. Once the model is complete, it is possible to predict the impact of a cyber asset failure/degradation as the realization of each IF...THEN statement, tracing the potential impact back "up" to key Tasks and Objectives, as shown in Figure 3.

Government Interest and Use

Government interest in identifying, prioritizing, and protecting critical elements of the national infrastructure crosses all departments and stems from Homeland Security Presidential Directive 7 (HSPD-7) [8]. Preserving the government's ability to perform essential missions and deliver essential services is among the key policy tenets in HSPD-7. These tenets were carried into the National Infrastructure Protection Plan (NIPP) developed by the

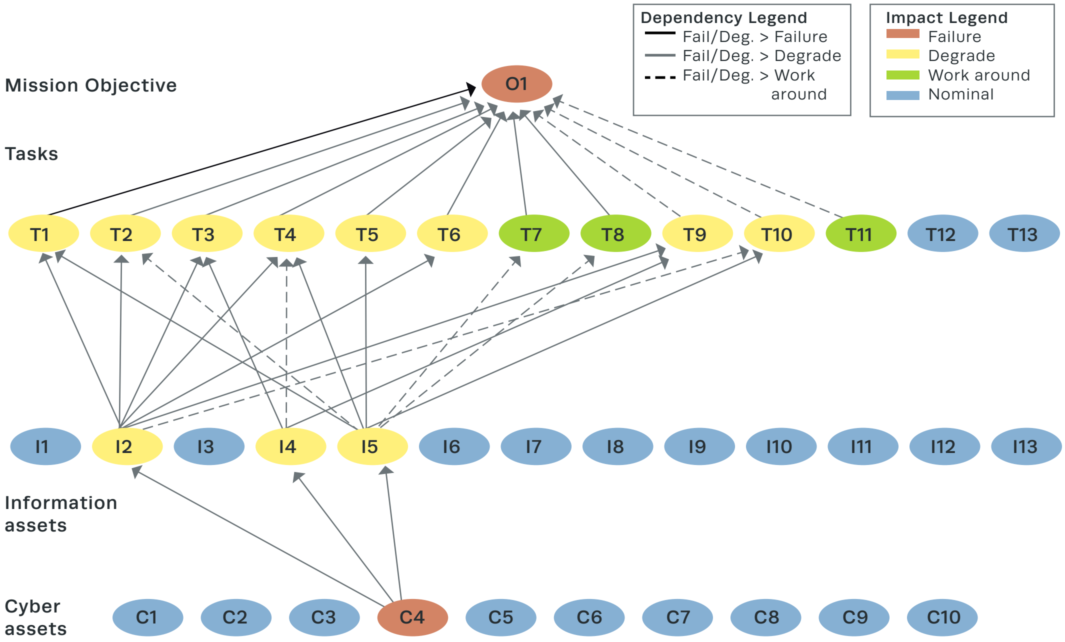


Figure 3. Predicted Impact of Cyber Asset Failure

Department of Homeland Security (DHS) [9]. The NIPP outlines a risk management strategy that allows for choices of risk management methodologies so long as they meet certain criteria. Among the Core Criteria for making consequence or impact assessments is "mission disruption." The special importance of mission disruption is clear in the NIPP, which states, "Mission disruption is an area of strong NIPP partner interest for collaborative development of the appropriate metrics to help quantify and compare different types of losses. While development is ongoing, qualitative descriptions of the consequences [of loss] are a sufficient goal."

Within the DoD, the Defense Critical Infrastructure Protection (DCIP) Program called for in DODD 3020.40 [10] and described in DODI 3020.45 [11] requires a mission-based criticality assessment of assets supporting DoD missions. The nine-step Critical Asset Identification Process (CAIP) set forth in DoDM 3020.45 V1 [12] requires Mission Owners and Asset Owners to jointly determine what is mission critical, based on performance criteria for missions and mission-essential tasks. CJA provides a useful framework for conducting this analysis, using Information Assets as the link between Tasks and the Cyber Assets that support them. Following the CJA with a TSA to determine risks and a RRA to identify possible risk mitigations is consistent with the guidance in these DoD documents.

Best Practices and Lessons Learned

The overarching goal. The general goal of a CJA is to make the adversary's job both more "difficult" (more costly and more time-consuming—hence more "expensive") and more risky.

Timing is critical. Generally the more enlightening insights come from using the dependency map methodology to evaluate system designs (and any alternatives or "design trades") after the system design starts to take shape. Some efforts have been made to perform a CJA early in an acquisition effort to identify mission-critical functions and mission-critical data. This can help identify information to protect at rest and in transit, where this information might be either a critical input or a computational "product" of the designed system.

Include all tasks needed to achieve the mission objectives. This means looking beyond those tasks that are on the critical path in a mission thread. What are the security-related tasks? What are the logistics-related tasks, or other support-related tasks? Excluding such tasks overlooks what must be done to ensure continued mission capability. An honest assessment of mission dependency on security-related, and other supporting tasks will ensure that the components supporting those tasks receive due attention.

Remember the supporting actors. Cyber assets that perform mission-critical functions are not the only crown jewels in a system. Any system components that have unmediated access to crown jewels, or provide protection for crown jewels, or enable crown jewels to perform their

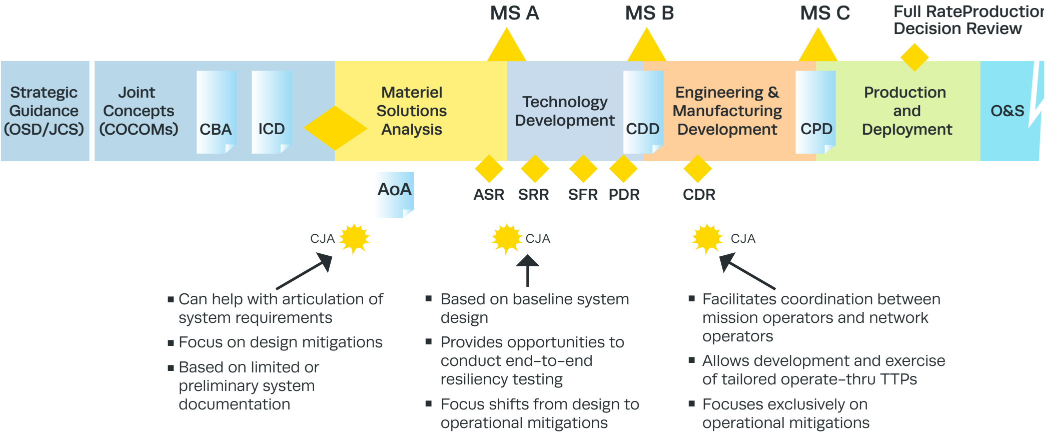

MS A
MS B
MS C
Full RateProductior
Decision Review
Strategic Guidance (OSD/JCS)
Joint Concepts (COCOMs)
CBA
ICD
Materiel Solutions Analysis
Technology Development
CDD
Engineering & Manufacturing Development
CPD
Production and Deployment
O&S
AoA
ASR
SRR
SFR
PDR
CDR
CJA
CJA
CJA
Can help with articulation of system requirements
Focus on design mitigations
Based on limited or preliminary system documentation
Based on baseline system design
Provides opportunities to conduct end-to-end resiliency testing
Focus shifts from design to operational mitigations
Facilitates coordination between mission operators and network operators
Allows development and exercise of tailored operate-thru TTPs
Focuses exclusively on operational mitigations

Figure 4. Alternate Time Frames for Performing CJA During a System Life Cycle

critical functions must themselves be considered for crown jewel status. Identifying them requires an understanding of the system architecture and the overall system functions, and the analysis will not be complete unless this step is performed. Forthcoming guidance in the Defense Acquisition Guidebook will address this point in the section on Criticality Analyses for Program Protection Planning.

The devil's in the details. System design details influence "criticality" in ways that developers—not operators—will more readily understand, so identifying key system accounts, critical files, and other critical assets will require technical insights from the development team (as depicted in the bottom two rows of Figures 2 and 3). Deciding which cyber assets are most important to "protect" (by close monitoring, using redundancy or other measures to more quickly restore these critical assets) is based on the insights provided by the dependency map "linkage" to the Tasks and Mission Objective. CJA can provide insight into which nodes to protect, where to apply intrusion detection, whether anti-tamper software and hardware are needed, and where and how to apply them.

Remember to factor in changing priorities. When circumstances require operators to "fight through" a cyber-attack, other considerations will also shape the CJA—such as putting priority into maintaining scenario-specific, "minimum essential" functions, or allowing for secure reconstitution from a pristine, protected software image that can be quickly patched to a current state and loaded with the latest data.

A life-cycle view of CJA. Figure 4 illustrates where CJAs might be performed at different points along a System Life Cycle (SLC) and describes the purposes they would serve. A CJA can be initiated at Milestone B and updated throughout the program, but it could certainly be started at a later phase if this analysis was not performed sooner.

Needs and resources—The big drivers. The choice of CJA method depends on the needs and resources of the requesting organization. If an organization's mission is clearly defined, with a modest number of tasks to accomplish their broader mission objectives, their crown jewels may be readily apparent. In some cases, they will have defined specific use cases; or in user parlance, "mission threads." For organizations that support many mission objectives and/or have complex and overlapping mission dependencies on IT, it is useful to employ the dependency map techniques of CJA. Where an even more detailed examination is necessary—based on complex ripple effects from IT failures or temporal effects during overlapping operations—a CMIA [13] approach offers the necessary capabilities to address these challenges.

References and Resources

1. Cyber Threat Susceptibility Assessment, Systems Engineering Guide.
2. Cyber Risk Remediation Analysis, Systems Engineering Guide.
3. Goldman, H., October 2010, Building Secure, Resilient Architectures for Cyber Mission Assurance, The MITRE Corporation.
4. Watters, J., S. Morrissey, D. Bodeau, and A. Powers, October 2009, The Risk-to-Mission Assessment Process (RiskMAP): A Sensitivity Analysis and an Extension to Treat Confidentiality Issues, The MITRE Corporation.
5. RiskMAP, MITREpedia article.
6. Musman, S., A. Temin, M. Tanner, D. Fox, B. Pridemore, July 2009, Evaluating the Impact of Cyber Attacks on Missions, The MITRE Corporation.
7. MITRE Institute Course TSV-418, Crown Jewels Analysis Using Dependency Maps.
8. Bush, President George W., December 2003, Critical Infrastructure Identification, Prioritization, and Protection. Homeland Security Presidential Directive/HSPD-7. The White House.
9. National Infrastructure Protection Plan, 2009, Department of Homeland Security.
10. DoD Policy and Responsibilities for Critical Infrastructure, DoD Directive 3020.40, Change 1. July 2010, Department of Defense.
11. Defense Critical Infrastructure Program (DCIP) Management, DoD Instruction 3020.45. April 2008, Department of Defense.
12. Defense Critical Infrastructure Program (DCIP): DoD Mission-Based Critical Asset Identification Process (CAIP), DoD Manual 3020.45 V1. October 2008, Department of Defense.

13. Musman, S., M. Tanner, A. Temin, E. Elsaesser, L. Loren, 2011, “A Systems Engineering Approach for Crown Jewels Estimation and Mission Assurance Decision Making,” *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security*, Paris, France, April 11–15, 2011.

Additional References and Resources

Cyber Operations Center, MITREpedia.

Hastings, G., L. Montella, and J. Watters, MITRE Crown Jewels Analysis Process, April 8, 2009.

Mission Assurance Against Cyber Attacks SharePoint site.

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Cyber Threat Susceptibility Assessment

Definition: *Cyber Threat Susceptibility Assessment (TSA) is a methodology for evaluating the susceptibility of a system to cyber-attack. TSA quantitatively assesses a system's [in]ability to resist cyber-attack over a range of cataloged attack Tactics, Techniques, and Procedures (TTPs) associated with the Advanced Persistent Threat (APT). In the Mission Assurance Engineering (MAE) methodology, cyber TSA is a follow-on to Crown Jewel Analysis (CJA), and a prerequisite to cyber Risk Remediation Analysis (RRA).*

Keywords: *advanced persistent threat, APT, cyber attack, MAE, mission assurance engineering, risk remediation, Threat Susceptibility Matrix, TTP*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of Mission Assurance Engineering (MAE) in the systems acquisition life cycle and its constituent activities, including cyber Threat Susceptibility Analysis (TSA). The MAE methodology has application throughout the system life cycle. MITRE SEs are expected to know the context(s) in which this methodology can be applied. MITRE SEs are also expected to help establish the scope of the assessment and set sponsor expectations regarding deliverables and schedules.

Introduction and Background

The MAE process framework (Figure 1 in the preceding article, “Crown Jewels Analysis”) provides an analytical approach to:

- Identify the cyber assets most critical to mission accomplishment (the “crown jewels” of a Crown Jewels Analysis).
- Understand the threats and associated risks to those assets (accomplished via a subsequent cyber Threat Susceptibility Analysis [TSA]).
- Select mitigation measures to prevent and/or fight through attacks (cyber Risk Remediation Analysis [RRA] is used to identify recommended mitigation measures).
- The MAE process framework provides a common repeatable risk management process that is part of building secure and resilient systems [1].

Cyber threat susceptibility analysis (TSA) is an MAE activity that quantitatively assesses a system’s [in]ability to resist cyber-attack over a range of adversary Tactics, Techniques, and Procedures (TTPs). A TSA assessment produces a Threat Susceptibility Matrix, which provides a ranked list of TTPs that cyber assets are susceptible to. This matrix is used in a follow-on MAE activity called cyber Risk Remediation Analysis, which develops recommendations for how to mitigate cyber TTP risk.

This article focuses on what we know today about cyber TSA, a fast-moving discipline. The concepts and methods described are evolving and will continue to mature as more experience is gained in applying this discipline.

The first step in a cyber TSA assessment is to establish the scope of the evaluation. Assessment scope is characterized in terms of:

- The set of assets being evaluated
- The range of attack TTPs being considered
- The types of adversaries

When TSA is conducted as a follow-on to a Crown Jewel Analysis (CJA), the set of system assets within the scope of the assessment may include identified crown jewel cyber assets (i.e., cyber assets whose compromise would seriously impair mission capability or readiness.) If the TSA is being conducted independently or in the absence of the CJA, the list of cyber assets may be arbitrarily selected or may include a presumptive list of crown jewel cyber assets.

The range of attacks considered in a TSA assessment may include but is not limited to cyber, electronic warfare (EW), and supply chain. A cyber attack targets an enterprise’s use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; destroying the integrity of the data; or stealing controlled information. Electronic warfare refers to military action involving the use of electromagnetic and directed energy to control the electromagnetic spectrum or to attack the enemy. Supply chain attacks allow the adversary to use implants or other vulnerabilities inserted prior to installation in order to infiltrate or exfiltrate data or to manipulate information technology hardware, software, operating systems, peripherals (information technology products), or services at any point during the life cycle. The Advanced Persistent Threat (APT) refers to adversaries, typically nation-states, capable of mounting sophisticated attacks in each of these areas.

Types of adversaries considered in a TSA assessment may include external adversaries, insiders, and trusted insiders. The distinctions among the types are fuzzy, but relate to the adversary's proximity to the targeted system. A security perimeter separates an external adversary from an internal adversary (i.e., an insider). This perimeter can take the form of a firewall, a DMZ, a locked door, and so on. Once the security perimeter is breached, however, the external adversary has gained insider access. Similarly, an insider is distinguished from a trusted insider by the level of access granted (i.e., a trusted insider may have physical or administrative access that an unprivileged user does not). Enforcing the principle of least privilege separates insiders from trusted insiders, who may have opportunities to apply a wider range of attack TTPs than insiders or external adversaries. The scope of a TSA assessment may include or exclude each of these types of adversaries.

Once the scope of the TSA assessment is established, the next step is to evaluate the cyber asset's architecture, technology, and security capabilities against a cataloged set of TTPs. Unclassified, open source TTP catalogs used in TSA assessments include MITRE-hosted resources such as:

- Common Attack Pattern Enumeration and Classification (CAPEC)—A compilation of cyber-attack patterns that describe common methods for exploiting software derived from specific real-world incidents [2]. In this context, the terms "attack TTP" and "attack pattern" are synonymous.
- Common Weakness Enumeration (CWE)—A catalog of defined software weaknesses that attack TTPs may exploit [3].
- Common Vulnerabilities and Exposures (CVE)—An open-source dictionary of publicly known information security vulnerabilities and exposures [4].

This initial set of TTPs undergoes a narrowing process to eliminate TTPs considered implausible. Several factors can make a TTP an implausible method of cyber-attack. Many TTPs have prerequisites or conditions that must hold true in order for that TTP to be effective. A prerequisite for a structured query language (SQL) injection attack, for example, is that the system must include a SQL database. Weak passwords is one condition that must hold true in order for an adversary to successfully conduct brute force password attacks. Many candidate attack TTPs may be eliminated because of missing prerequisites.

It is also possible to eliminate candidate attack TTPs by making assumptions about the system's security posture. For example, DoD systems undergo the Defense Information Assurance Certification and Accreditation (DIACAP) process to verify that all required security controls are implemented. One set of security controls requires that the system's configuration be hardened using Defense Information Systems Agency published Security Technical Implementation Guides (STIGs). Certain attack TTPs may not be plausible for systems that have been hardened in accordance with these STIGs.

Candidate attack TTPs that cannot be eliminated may be ranked using a scoring model that assesses the risk associated with each TTP relative to other plausible TTPs considered in the assessment. This ranking helps set priorities on where to apply security measures to reduce the system's susceptibility to cyber-attack. The default TTP scoring model spreadsheet is illustrated in Table 1.

Table 1. Default TTP Risk Scoring Model

Factor range	**1**	**2**	**3**	**4**	**5**
Proximity What proximity would an adversary need in order to apply this TTP?	No physical or network access required	Protocol access through DMZ and firewall	User account to target system (no admin access)	Admin access to target system	Physical access to target system
Locality How localized are the effects posed by this TTP?	Isolated to single unit	Single unit and supporting network	External networks potentially impacted	All units in theater or region	All units globally and associated infrastructure
Recovery Time How long would it take to recover from this TT P once the attack was detected?	< 10 hours	20 hours	30 hours	40 hours	>50 hours
Restoration Costs What is the estimated cost to restore or replace affected cyber asset?	< $10K	$25K	$50K	$75K	>$100K
Impact How serious an impact is loss of data confidentiality resulting from successful application of this TTP?	No impact from TTP	Minimal impact	Limited impact requiring some remediation	Remediation activities detailed in COOP	COOP remediation activities routinely exercised

Impact How serious in impact is loss of data integrity resulting from successful application of this TTP?	No impact from TTP	Minimal impact	Limited impact requiring some remediation	Remediation activities detailed in COOP	COOP remediation activities routinely exercised
Impact How serious an impact is loss of system availability resulting from successful application of this TTP?	No impact from TTP	Minimal impact	Limited impact requiring some remediation	Remediation activities detailed in COOP	COOP remediation activities routinely exercised
Prior Use Is there evidence of this TTP in the MITRE Threat DB?	No evidence of TTP use in MTDB	Evidence of TTP use possible	Confirmed evidence of TTP use in MTDB	Frequent use of TTP reported in MTDB	Widespread use of TTP reported in MTDB
Required Skills What level of skill or specific knowledge is required by the adversary to apply this TTP?	No specific skills required	Generic technical skills	Some knowledge of targeted system	Detailed knowledge of targeted system	Knowledge of both mission and targeted system
Required Resources Would resources be required or consumed in order to apply this TTP?	No resources required	Minimal resources required	Some resources required	Significant resources required	Resources required and consumed
Stealth How detectable is the TTP when it is applied?	Not detectable	Detection possible with specialized monitoring	Detection likely with specialized monitoring	Detection likely with routine monitoring	TTP obvious without monitoring
Attribution Would residual evidence left behind by this TTP lead to attribution?	No residual evidence	Some residual evidence, attribution unlikely	Attribution possible from characteristics of the TTP	Some or similar TTPs previously attributed	Signature attack TTP used by adversary

The default TTP scoring model assesses TTP risk based on twelve criteria, including impact, restoration costs, down time, level of sophistication, likelihood for attribution, and so on. This list of criteria, which has evolved over time, may be tailored for use in a given assessment. Use of the same scoring model provides a common context for comparing and ranking TTPs based on relative risk. TTP risk scores derived using different scoring models are not comparable.

A uniform range of values [1–5] is assigned to each criterion. For criteria such as impact, a higher value results in a higher TTP risk score. These criteria appear in blue in the scoring model spreadsheet. For adversary level of sophistication criteria, such as required skills and required resources, a higher value results in a lower TTP risk score. These criteria appear in red

Table 2. Sample Threat Susceptibility Matrix

TTP ID	Risk Score	Cyber Asset #1			Cyber Asset #2		
		External	Insider	Trusted Insider	External	Insider	Trusted Insider
T000017	4.4		4.4	4.4		4.3	4.3
T000030	4.2		4.1	4.1		4.1	4.1
T000039	3.6	3.6	3.6			3.6	
T000041	3.2	3.2	3.2		3.2	3.2	
T000053	3.0						3.0
T000064	2.9				2.9		
T000086	2.6				2.6	2.6	2.6
T000127	2.6				2.6		
T000018	2.3					2.3	2.3
T000022	2.3	2.3	2.3	2.3	2.3	2.3	2.3
T000023	2.3	2.3	2.3		2.3	2.3	
T000029	2.2	2.2	2.2		2.2	2.2	
T000048	2.0			2.0			
T000054	1.9				1.9	1.9	
T000063	1.6				1.6	1.6	
T000065	1.3	1.3					
Aggregate Susceptibility		14.9	22.1	12.8	21.6	30.4	18.6
		49.8			70.6		

in the scoring model spreadsheet. In the threat model from which this scoring model derives, it is assumed that a high degree of adversary sophistication required to successfully execute a TTP reduces the overall likelihood of occurrence, leading to a lower overall risk score.

In the default TTP scoring model, different criteria can have different weightings. Some criteria may be more significant to the overall risk score than others. For a system that processes classified data, for example, a higher weighting is assigned to loss of confidentiality than for a system that processes unclassified data. TTP risk scores are calculated based on the criteria value assignments and associated criteria weightings. In the default scoring model, this calculation yields a TTP risk score in the range [1–5], with the value 5 signifying a TTP that poses the greatest risk.

A TSA assessment produces a Threat Susceptibility Matrix, which lists plausible attack TTPs ranked by decreasing risk score, and their mapping to cyber assets as a function of adversary type. The Threat Susceptibility Matrix also tabulates TTP risk scores to provide an overall assessment of aggregate susceptibility to cyber-attack for each cyber asset considered in the assessment. This matrix is used in the follow-on cyber risk remediation analysis (RRA) to identify countermeasures that effectively mitigate TTP susceptibilities. For further information on cyber RRA, see the companion article under this same topic. A sample Threat Susceptibility Matrix is illustrated in Table 2.

The sample Threat Susceptibility Matrix in Table 2 evaluates two cyber assets over a range of sixteen attack TTPs that are scored using the default TTP scoring model from Table 1. If a cyber asset is susceptible to a TTP, its risk score is transferred to that cyber asset. At the bottom of the matrix, aggregate susceptibility is tabulated for each cyber asset and adversary type. In this example, Cyber Asset #2 is more susceptible than Cyber Asset #1.

TTPs are "binned" into risk categories based on risk score, as follows:

- TTPs in the range [4.0–5.0] pose serious risk and appear in red.
- TTPs in the range [2.5–3.9] pose moderate risk and appear in yellow.
- TTPs in the range [1.0–2.4] pose minimal risk and appear in blue.

Government Interest and Use

TSA has been applied to sponsor systems in various forms for a number of years. Before 2010, TSA assessments used a loosely defined, non-rigorous, and undocumented methodology. In 2010, a formal methodology for conducting TSA assessments was developed by MITRE, which has subsequently been applied to Army, Navy, and Air Force programs [5]. The methodology outlined above reflects this TSA approach.

Best Practices and Lessons Learned

Timing is critical. TSA may not be well suited to all phases of acquisition programs. For example, the Threat Susceptibility Matrix cannot be constructed without knowledge of the cyber assets that make up the system. The identification of cyber assets is derived from the system's allocated baseline, which may not be fully defined prior to PDR.

Assume the adversary can gain access. Mission Assurance Engineering (MAE) is based on the assumption that APT adversaries are able to successfully penetrate a system's security perimeter and gain persistent access. TSA's focus on the Insider or Trusted Insider relates to adversary proximity and in no way reflects on IT staff loyalty or ability.

TSA of non–crown jewel assets—the value proposition. Although a Crown Jewel Analysis (CJA) identifies cyber assets of greatest importance to mission success, it does not identify the cyber assets that are most susceptible to attack. There is value in scoping a TSA assessment to evaluate non–crown jewel cyber assets, especially those whose compromise would give an attacker a path to any crown jewel assets.

Importance of documented rationale as context for future efforts. It is important to record the rationale for eliminating candidate attack TTPs from consideration, including assumptions made regarding the system's security posture or Security Technical Implementation Guide (STIG) compliance. The rationale provides context in follow-on MAE activities such as Threat Remediation Engineering (TRE).

More than one cyber risk assessment methodology. Several methodologies similar to TSA assess risk to cyber assets, including Microsoft's DREAD [6] and the National Security Agency's MORDA [7]. Each methodology functions by assessing cyber assets using a defined set of evaluation criteria. The criteria used in this article's default TTP scoring model are representative of criteria used by these other methodologies and can be tailored to meet the needs of the program.

Pathological scores and what to do about them. Certain "pathological" TTP scoring modes may reflect situations where more information about a cyber asset is required, evaluation criteria need to be revised, or the assigned range of values is either too narrow or too wide. When tailoring the scoring model to address this, it is necessary to go back and rescore all TTPs using the updated model. Otherwise, a single scoring model is not being used and there is no basis for comparing TTP risk scores in the assessment.

Variation on the TTP risk scoring theme. One variation on the TTP risk score calculation is to compute low and high TTP risk scores based on a range of values for each evaluation criteria. This approach produces risk scoring that reflects best case and worst case assumptions.

The need for remediation. A cyber TSA provides no recommendations on how to respond to cyber risk. Cyber Risk Remediation Analysis (RRA) is the core MAE portfolio activity used to identify risk mitigation alternatives. Threat Assessment and Remediation Analysis (TARA) [8] is the MAE portfolio practice that combines Cyber TSA with RRA. Sponsors seeking to identify, assess, and mitigate cyber risks are encouraged to consider a TARA assessment.

References and Resources

1. Goldman, H., October 2010, Building Secure, Resilient Architectures for Cyber Mission Assurance, The MITRE Corporation.
2. Common Attack Pattern Enumeration and Classification (CAPEC).
3. Common Weakness Enumeration (CWE).
4. Common Vulnerabilities and Exposures (CVE).
5. Wynn, J., and L. Montella, October 2010, "Cyber Threat Susceptibility Analysis (TSA) Methodology," Ver. 2.0, MITRE Technical Report 100379.
6. Meier, J. D., A. Mackman, et al., June 2003, Microsoft Patterns and Practices, Chapter 3, Threat Modeling.

7. Buckshaw, D., G. Parnell, et al., 2005, "Mission Oriented Risk and Design Analysis of Critical Information Systems (MORDA)," *Military Operations Research*, Vol. 10, No. 2, pp. 19–38.

8. Wynn, J., et al., October 2011, "Threat Assessment and Remediation Analysis (TARA)", Ver. 1.4, MITRE Technical Report 110176, The MITRE Corporation.

Definition: *Cyber Risk Remediation Analysis (RRA) is a methodology for selecting countermeasures to reduce a cyber-asset's susceptibility to cyber-attack over a range of attack Tactics, Techniques, and Procedures (TTPs) associated with the Advanced Persistent Threat (APT). In the Mission Assurance Engineering (MAE) methodology, RRA is a follow-on to cyber Threat Susceptibility Analysis (TSA) and provides recommendations to sponsors seeking to reduce susceptibility to cyber-attack.*

Keywords: *advanced persistent threat, APT, CM, countermeasure, cyber-attack, MAE, mission assurance engineering, risk Remediation, RRA, threat susceptibility, TSA, TTP, recommendation, utility-cost ratio, U/C ratio*

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Cyber Risk Remediation Analysis

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of Mission Assurance Engineering (MAE) in the systems acquisition life cycle and its constituent activities, including cyber Risk Remediation Analysis (RRA). The MAE methodology has application throughout the system life cycle. MITRE SEs are expected to know the context(s) in which this methodology can be applied. They are also expected to help establish the scope of the assessment and set sponsor expectations regarding deliverables and schedules.

Introduction and Background

The MAE process framework (Figure 1 in the “Crown Jewels Analysis” article) provides an analytical approach to:

- Identify the cyber-assets most critical to mission accomplishment (the “crown jewels” of a Crown Jewels Analysis).
- Understand the threats and associated risks to those assets (accomplished via a subsequent cyber Threat Susceptibility Analysis [TSA]).
- Select mitigation measures to prevent and/or fight through attacks (cyber Risk Remediation Analysis [RRA] is used to identify recommended mitigation measures).

The MAE process framework provides a common repeatable risk management process that is part of building secure and resilient systems [1].

Cyber risk remediation analysis (RRA) is the final step in the MAE process framework. It is a methodology for selecting countermeasures (CMs) to reduce a cyber-asset’s susceptibility to cyber-attack over a range of tactics, techniques, and procedures (TTPs) associated with the APT. A CM is defined as an action, device, procedure, or technique that opposes or counters a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by detecting and reporting it so that corrective action can be taken [2]. The selection of CMs is governed by the system life cycle of the cyber-asset being evaluated. Recommended CMs are those judged to be effective at mitigating TTPs to which a cyber-asset may be susceptible. CMs cover a broad spectrum, including changes to requirements, system design, testing, deployment configuration, and/or operating procedures.

This article focuses on what we know today about cyber RRA, a fast-moving branch of systems engineering. The concepts and methods described are evolving and will continue to mature as more experience is gained in applying this discipline. Please revisit this article for additional insights as the community’s collective knowledge builds.

The Advanced Persistent Threat (APT) refers to an adversary with sophisticated levels of expertise and significant resources that can apply multiple, different attack vectors to achieve its objectives, which include the establishment of footholds within the information technology infrastructure of an organization to continually exfiltrate information and/or undermine or impede critical aspects of a mission, program, or organization, or to place itself in a position to do so in the future [3]. The APT pursues its objectives over an extended period of time, adapts to a defender’s efforts to resist it, and maintains the level of interaction needed to execute its objectives.

Cyber RRA is a follow-on to a cyber Threat Susceptibility Analysis (TSA), which produces a Threat Susceptibility Matrix that ranks TTPs and maps them to cyber-assets. In a TSA assessment, a scoring model spreadsheet may be used to rank TTPs on a risk scale of [1–5], with 1 representing very low risk and 5 representing very high risk. Factors used in the TTP risk scoring spreadsheet can vary from one assessment to the next, but must be uniformly applied across all TTPs evaluated in an assessment to ensure consistent ranking. This scoring tool can be tailored, (e.g., add or remove criteria, modify weightings) or even replaced to meet a program's needs. The interested reader is referred to [4, 5] for details on TSA and the default TTP risk scoring model.

The first step in cyber RRA is to use the Threat Susceptibility Matrix to identify which TTPs to mitigate for each cyber-asset. There are several strategies for performing this selection. One strategy is to focus only on the highest risk TTPs of each cyber-asset. Another strategy is to focus on the cyber-asset(s) that have the highest aggregate susceptibility. Aggregate susceptibility is calculated for each cyber-asset and category of threat actor by summing the risk scores of the mapped TTPs. Note that these calculations use rounded risk scores and will be subject to rounding errors. A third strategy is for RRA to focus exclusively on crown jewel cyber-assets. A hybrid approach might select high-risk TTPs for the crown jewel cyber-assets with the highest aggregate susceptibility. Whatever strategy is used, the result will be a list of TTPs for each cyber-asset assessed.

Table 1 in the preceding article, “Cyber Threat Susceptibility Assessment,” provided a notional example of a Threat Susceptibility Matrix for two cyber-assets: cyber-asset #1 and cyber-asset #2. In this example, both assets are judged to be essentially equally susceptible to high-risk TTPs T000017 and T000030. Overall, cyber-asset #2 is more susceptible than cyber-asset #1 to a range of TTPs, as reflected by its higher aggregate susceptible scores. The color coding indicates the relative severity of the threat.

Because different cyber-assets are susceptible to different TTPs, cyber RRAs are conducted separately for each cyber-asset. The RRA uses a mapping table that associates TTPs with CMs. A sample TTP/CM mapping table is illustrated in Table 2.

Each row in a TTP/CM mapping table corresponds to a countermeasure and each column corresponds to a TTP. A CM to TTP mapping is characterized by the mitigation effectiveness of the CM over a range of criteria: detect, neutralize, limit, and recover. Detect CMs serve to identify or uncover the action or presence of a TTP. Neutralize CMs stop or prevent execution of a TTP. Limit CMs serve to reduce or constrain the risk associated with a TTP, either by lessening the severity or likelihood. Recovery CMs facilitate recovery from attack. A given CM may be highly effective at detecting a certain TTP, moderately effective at neutralizing or limiting its impact, but provide no mitigation value in recovering from its effects. A 2-character notation is used to denote mitigation effectiveness within the mapping table, where the first character signifies the type of mitigation from the list: (N)eutralize, (D)etect, (L)imit, and (R)ecover, and the second character represents the degree of effectiveness from the list: (L)ow, (M)edium, (H)igh, and (V)ery high. The value NH represents Neutralize-High mitigation effectiveness, while the value DM represents Detect-Medium mitigation effectiveness.

Table 2. TTP/CM Mapping Table

CM ID	Mitigation Effectiveness (by TTP ID)							
	T000017	T000030	T000039	T000041	T000053	T000064	T000086	T000127
C000039						NM		
C000045		NH	NH					
C000047			NH					
C000058			NH					
C000067	DL, NM							
C000073				LM				
C000083	LH,NH							
C000086				LM				
C000096				NM				
C000097				DM, NM				
C000110				LM, NL				
C000113				NM				
C000121							DM, NM	
C000122				NM				
C000124			LM					
C000126					LM			
C000129							NM	
C000133						NM		
C000144			NH					
C000145	NH		NH					
C000147		NM						
C000159						NM		
C000164		NM						
C000165				LH				
C000173		NM						
C000187								LM
C000188		NM						

The RRA seeks to identify a list of CMs that mitigate a list of TTPs based on this mapping table. This list is optimal if it identifies CMs that are highly effective at mitigating most or all of the listed TTPs at a minimal cost. One key assumption made with this approach is that CMs can be applied in combination to achieve greater mitigation effectiveness than they would individually, which is the basis for the onion model of security.

To identify an optimal list of CMs, it is first necessary to assess the relative merit of each CM. One approach, detailed below, is to calculate the utility/cost (U/C) ratio for each CM. A U/C ratio is a “bang-for-buck” valuation of a CM derived from its estimated utility and cost.

With the default scoring model, CM utility is estimated by assigning a numeric score to each mitigation effectiveness value and multiplying by the number of mappings containing that mitigation effectiveness value across the list of TTPs being assessed. Once U/C ratios are calculated for each CM, the CM ranking table is sorted by descending U/C ratios. This approach for calculating U/C rations is depicted in Table 3.

The next step is to walk through the ranking table to identify sets of CMs that mitigate the list of TTPs, starting at the top and working down. Ordering CMs in the ranking table by descending U/C ratio facilitates early identification of optimal solutions. When a combination

Table 3. CM Ranking Table Example

CM ID	Neutralize			Limit		Detect		CM Merit Scoring		
	NH = 9	NM = 7	NL = 5	LH = 7	LM = 5	DM = 3	DL = 1	Utility	Cost	U/C Ratio
C00159		T000064						7	1	7.0
C00164		T00030						7	1	7.0
C00165				T000041				7	1	7.0
C00173		T000030						7	1	7.0
C00188		T000030						7	1	7.0
C00045	T000030, T000039							18	3	6.0
C00145	T000039, T000017							18	3	6.0
C00083	T000017			T000017				16	3	5.3
C00073					T000041			5	1	5.0
C00067		T000017					T000017	8	2	4.0
C00096		T000041						7	2	3.5
C00113		T000041						7	2	3.5
C00133		T000064						7	2	3.5
C00097		T000041				T000041		10	3	3.3
C00110			T000041		T000041			10	3	3.3
C00047	T000039							9	3	3.0
C00058	T000039							9	3	3.0
C00144	T000039							9	3	3.0
C00086					T000041			5	2	2.5
C00121		T000086				T000086		10	4	2.5
C00124					T000039			5	2	2.5
C00126					T000053			5	2	2.5
C00187					T000127			5	2	2.5
C00039		T000064						7	3	2.3
C00122		T000041						7	3	2.3
C00129		T000086						7	3	2.3
C00147		T000030						7	3	2.3

of CMs is identified that provides mitigation value over the range of TTPs, a solution effectiveness table is constructed to illustrate the coverage provided and to calculate the relative cost for that solution. These solution effectiveness tables are used to compare alternative solutions. Tables 4 and 5 represent two alternative solutions that provide roughly equivalent mitigation over the same list of TTPs but at different costs.

In addition to providing a basis for comparing alternative solutions, solution effectiveness tables document the mapping between each TTP and the CMs that provide mitigation value. They can be used to identify cases where mitigation redundancies or coverage gaps exist. For example, additional countermeasures may be advisable for T000053 and T000127 in both solution alternatives above.

Table 4. Solution Effectiveness Table Example #1

CM ID	Cost	Mitigation Effectiveness (by TTP ID)							
		T000017	T000030	T000039	T000041	T000053	T000064	T000086	T000127
C000045	3		NH	NH					
C000083	3	LH,NH							
C000126	2					LM			
C000129	3							NM	
C000165	1				LH				
C000187	2								LM
C000159	1						NM		
Total	15	NH	NH	NH	LH	LM	NM	NM	LM

Table 5. Solution Effectiveness Table Example #2

CM ID	Cost	Mitigation Effectiveness (by TTP ID)							
		T000017	T000030	T000039	T000041	T000053	T000064	T000086	T000127
C000058	3		NH	NH					
C000067	2	DL,NM							
C000096	2				NM				
C000133	2						NM		
C000121	4							DM,NM	
C000126	2					LM			
C000147	2		NM						
C000187	2								LM
Total	20	NM	NH	NH	NM	LM	NM	NM	LM

The final step is to translate the list of CMs reflected by the lowest cost solution into well-formed recommendations. A well-formed recommendation includes three pieces of information:

- The action, device, procedure, or technique recommended (i.e., which CM to apply)
- The reason why the CM is required (i.e., the TTPs that it mitigates)
- The implication or effect if the CM is not applied (i.e., the potential impact to mission capability resulting from compromise of the cyber asset)

A cyber RRA conducted as follow-on to a cyber TSA addresses the first two items above. To detail all three elements, however, a crown jewel analysis may be necessary in order to identify the range of mission impact(s) that result from compromise of a cyber asset.

Best Practices and Lessons Learned

Adapt RRA to satisfy program needs. The objective of an RRA assessment may not be to identify an optimal solution but instead to understand the range of mitigation alternatives and/or areas where gaps exist. In this context, the RRA deliverable would consist of TTP/CM mapping table data for a specified set of TTPs.

Use TARA to evaluate cyber risks and mitigations. Early assessments demonstrated that a TSA conducted without a follow-on RRA provides limited value to sponsors who seek to reduce susceptibility to cyber-attack. The MAE portfolio now combines cyber TSA and RRA into a single engineering practice called Threat Assessment and Remediation Analysis (TARA) [5].

Consider alternative scoring approaches. A variety of more sophisticated scoring models and approaches can be considered prior to conducting an assessment. The RRA approach does not mandate using U/C ratios or the default RRA scoring model; any approach for estimating CM merit may be used provided all CMs are assessed consistently.

Automate to manage large amounts of data. Large catalogs of TTPs and CMs produce very large TTP/CM mapping tables, which require automation to be processed effectively.

Evaluate security measures for operational systems. For deployed and operational systems, one optional step not discussed above is the evaluation of existing security measures to determine whether effective TTP mitigations have already been applied. In cases where such security measures are judged to be highly effective, it may be expedient to remove the TTP from the list of TTPs being evaluated for that cyber-asset.

Reduce the CM search space. The process used to enumerate the solution set of CMs can benefit from the application of heuristics that reduce the search space. The heuristic outlined previously is to rank CMs by U/C ratio in order to facilitate early identification of optimal (i.e., lowest cost) solutions. Other heuristics may also apply.

Crown jewel analysis is essential. A well-formed recommendation details risk-to-mission impact, which is needed to make informed decisions. A crown jewel analysis or other form of mission impact analysis is essential for that determination.

References and Resources

1. Goldman, H., October 2010, Building Secure, Resilient Architectures for Cyber Mission Assurance, The MITRE Corporation.
2. NIST Special Publication 800-39, March 2011, Integrated Enterprise-Wide Risk Management.
3. CNSS Instruction 4009, National Information Assurance (IA) Glossary, April 2010, Cyber Threat Susceptibility Analysis (TSA), Systems Engineering Guide.
4. Wynn, J., and L. Montella, October 2010, "Cyber Threat Susceptibility Analysis (TSA) Methodology," Ver. 2.0, MTR 100379, The MITRE Corporation.
5. Wynn, J., et al., October 2011, "Threat Assessment and Remediation Analysis (TARA)," Ver. 1.4, MTR 110176, The MITRE Corporation.

Definition: *A secure code review is a specialized task involving manual and/or automated review of an application's source code in an attempt to identify security-related weaknesses (flaws) in the code. A secure code review does not attempt to identify every issue in the code, but instead looks to provide insight into what types of problems exist and to help the developers of the application understand what classes of issues are present. The goal is to arm the developers with information to help them make the application's source code more sound and secure.*

Keywords: *code review, evaluation, secure code, secure code review, secure development, security, test, vulnerable software*

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Secure Code Review

MITRE SE Roles and Expectations: MITRE system engineers (SEs) often help our sponsors and customers formulate plans and policies for developing applications through all stages of the software development life cycle. Security has become a major point of emphasis and a key component within the larger area of mission assurance. Writing source code that is sound and secure is key in creating applications that withstand attack and function as intended in the face of a malicious adversary. As a consequence, MITRE SEs are expected to understand the rationale behind a secure code review and when such a review is appropriate. They are expected to understand where a secure code review fits into the software development life cycle and how it can be used most effectively to identify potential issues within the code. Finally, MITRE SEs are expected to understand how a secure code review is performed and what its strengths and limitations are.

Background

Application-level security is increasingly coming under fire. The Stuxnet worm [1] in 2010 was a high-profile example of how a malicious user can leverage an application vulnerability to subvert protection mechanisms and damage an end system. Verifying that applications correctly implement security mechanisms and do not contain vulnerabilities is critical to achieving mission assurance goals.

Compounding the problem are the facts that applications are becoming more interconnected and that flaws in one application often lead to exploitation of other applications. There is no unimportant application from a security point of view. Malicious users are eager to take advantage of any flaw in any application that enables them to achieve their goal.

Almost all software development life cycles include testing and validation, which is often accomplished as a code review by either a peer or an external entity. The review verifies that the application functions as expected and that required features have been implemented correctly. Code reviews are important and should still occur. However, an additional review with a focus solely on security should also be conducted.

A secure code review is a specialized task involving manual and/or automated review of an application's source code in an attempt to identify security-related weaknesses (flaws) in the code. It does not attempt to identify every issue in the code, but instead looks to provide insight into what types of security-related problems exist and help the application developer understand what classes of issues are present. A secure code review will not necessarily find every security flaw in an application, but it should arm developers with information to help make the application's source code more sound and secure.

The goal of a secure code review is to find and identify specific security-related flaws within the code that a malicious user could leverage to compromise confidentiality, integrity, and availability of the application. For example, an unchecked input to a buffer may enable a write past the end of the buffer and allow a malicious user to execute arbitrary code with escalated privileges. A secure code review looks to find these types of issues, notify development teams of them, and eventually result in fewer vulnerabilities in the application.

A secure code review is not a silver bullet, but it is a strong part of an overall risk mitigation program to protect an application.

Government Interest and Use

On October 29, 2010, The Defense Information Systems Agency (DISA) issued Version 3, Release 2, of the Application Security and Development Security Technical Implementation

Guide (STIG) [2]. This document "provides the guidance needed to promote the development, integration, and updating of secure applications" throughout their life cycles.

Department of Defense (DoD) Directive (DoDD) 8500.01E requires that all information assurance (IA) and IA-enabled information technology (IT) products incorporated into DoD information systems be configured in accordance with DoD-approved security configuration guidelines, and it tasks DISA to develop and provide security configuration guidance for IA and IA-enabled IT products in coordination with the Director, National Security Agency. The Application Security and Development STIG is provided under the authority of DoDD 8500.01E.

APP5080 within the Application Security and Development STIG mandates a secure code review before an application is released.

Focus of a Secure Code Review

A secure code review focuses on seven security mechanisms, or areas. An application that is weak in any area makes itself a target for a malicious user and increases the likelihood that the application will be used in an attack. A secure code review should inform the developers of the soundness of the source code in each of these areas:

- Authentication
- Authorization
- Session management
- Data validation
- Error handling
- Logging
- Encryption

Several weaknesses (flaws) can affect each of the preceding security mechanisms. Flaws in the handling of passwords often affect authentication. Flaws related to the type of information included in a message often affect error handling. Flaws in regular expressions often affect data validation.

The Common Weakness Enumeration [3] is a listing of the specific types of flaws that a secure code review looks for. "It serves as a common language for describing software security weaknesses, as a standard measuring stick for software security tools targeting these vulnerabilities, and as a baseline standard for weakness identification, mitigation, and prevention efforts."

Manual vs. Automated Review

A secure code review can be a manual or automated review, each with advantages and disadvantages. In a manual review, an analyst reviews the code line by line, looking for defects and security-related flaws. An automated review uses a tool to scan the code and report potential flaws.

Manual review is time-consuming and requires significant domain expertise to be done correctly. Often it takes years of experience to become efficient at manual code review. Even with experienced human analysis, errors in the review (missed and incorrect findings) are unavoidable. A proficient reviewer can get through about 3,000 lines of code a day, based on the experiences of the MITRE Secure Code Review Practice.

Automated review helps solve the problems associated with manual review. However, good automated review tools are expensive. Additionally, the technology behind automated tools is only effective at finding certain types of flaws. A single automated tool may be good at finding some issues but unable to detect others. Employing multiple automated tools can mitigate this problem but will still not uncover every issue. Automated tools also tend to produce false positives (reported findings that are not actually issues). Adjudicating false positives requires human intervention and takes time away from the development team.

The best approach for a secure code review is to understand the advantages and disadvantages of each method and to incorporate both as appropriate.

When to Perform a Secure Code Review

Security should be a focus throughout the entire development life cycle. Creating threat models during the design phase, educating developers on secure coding practices, and performing frequent peer reviews of code with security personnel involved will all help increase the overall quality of the code and reduce the number of issues reported (and hence that need to be fixed) by the secure code review.

However, a secure code review is best used toward the end of the source code development, when most or all functionality has been implemented. The reason for waiting until late in the development phase is that a secure code review is expensive and time-consuming. Performing it once toward the end of the development process helps mitigate cost.

Best Practices and Lessons Learned

Understand the developers' approach. Before starting a secure code review, talk to the developers and understand their approaches to mechanisms like authentication and data validation. Information gathered during this discussion can help jump-start the review and significantly decrease the time a reviewer spends trying to understand the code.

Use multiple techniques. If possible, use both manual and automated techniques for the review because each method will find things that the other doesn't. In addition, try to use more than one automated tool because the strengths of each differ and complement the others.

Do not assess level of risk. A secure code review should not attempt to make judgments about what is acceptable risk. The review team should report what it finds. The customer uses the program's approved risk assessment plan to assess risk and decide whether to accept it or not.

Focus on the big picture. When performing a manual review, resist trying to understand the details of every line of code. Instead, gain an understanding of what the code as a whole is doing and then focus the review on important areas, such as functions that handle login or interactions with a database. Leverage automated tools to get details on specific flaws.

Follow up on review points. After a review, hold a follow-up discussion with the development team to help them understand what the findings mean and how to address them.

Stick to the intent of the review. Secure code review is not penetration testing. Review teams should not be allowed to "pen-test" a running version of the code because it can bias the results by giving a false sense of completeness.

Limitations of a Secure Code Review

A secure code review is not a silver bullet, and performing such a review does not mean that every instance of a security flaw will be discovered. Rather it is one of many different types of activities that can help increase the quality of an application and reduce the number vulnerabilities in software, making it more difficult for a malicious user to exploit.

References and Resources

1. Wikipedia contributors, Wikipedia, The Free Encyclopedia, "Stuxnet," accessed February 14, 2011.
2. Application Security and Development Security Technical Implementation Guide, Defense Information Systems Agency, Ver. 3, Rel. 3.
3. The MITRE Corporation, Common Weakness Enumeration, cwe.mitre.org/.

Additional References and Resources

Department of Homeland Security, 2010, Software Assurance Pocket Guide Series: Software Security Testing.

Dowd, M., J. McDonald, and J. Schuh, 2007, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*, Addison-Wesley, ISBN 0-321-44442-6.

Viega, J., and G. McGraw, 2002, *Building Secure Software: How to Avoid Security Problems the Right Way*, Addison-Wesley, ISBN 0-201-72152-X.

Definition: *Supply Chain Risk Management (SCRM) is a discipline that addresses the threats and vulnerabilities of commercially acquired information and communications technologies within and used by government information and weapon systems. Through SCRM, systems engineers can minimize the risk to systems and their components obtained from sources that are not trusted or identifiable as well as those that provide inferior material or parts.*

Keywords: *advanced cyber threat, configuration management, emerging threat, materiel, program protection plan, risk management, supply chain, systems engineering process*

SYSTEMS ENGINEERING FOR MISSION ASSURANCE

Supply Chain Risk Management (SCRM)

MITRE SE Roles and Expectations: The expansion of the global economy, increased use of outsourcing, and development of open standards are some of the modern-day factors that present new challenges to the security of government systems. These factors have resulted in emerging threats and have made protection of the supply chain increasingly difficult [1]. All MITRE systems engineers (SEs) must understand these emerging threats and why SCRM is necessary to ensure the protection and viability of all government systems.

Why SCRM Is Important

The National Security Presidential Directive 54, Homeland Security Presidential Directive 23, and Defense Authorization Act 254 have made SCRM a national priority [2, 3]. Thus the Department of Defense (DoD), Department of Homeland Security, and other departments have begun to review and refine their SCRM practices and procedures. The goal of one of the DoD Comprehensive National Cyber Initiatives (CNCI) is to provide the U.S. government with a robust toolset of supply chain assurance defense-in-breadth and defense-in-depth methods and techniques. The CNCI effort conducted a pilot program and produced a Key Practices Guide to provide SEs with key practices that can help manage supply chain risk. All SEs should become familiar with ongoing efforts within their sponsor's organization and materials like the Key Practices Guide. A summary of best practices follows.

Best Practices and Lessons Learned

Supply chain analysis. To determine the applicability of SCRM to a MITRE systems engineering project or initiative, MITRE SEs must comprehend or become educated on supply chain materiel management processes, the emerging threat, and the current supply chain challenges. This background will assist SEs in assessing which systems, components, software, organizational processes, and workforce issues have vulnerabilities or weaknesses that can be exploited.

The term "supply chain" has different meanings to commercial, government, and commercial entities. The military has extensive processes for structuring supplies (materiel management) to their units and organizations (see DoD 4140.1-R) [4]. Historically, the DoD has assessed the logistical tail of supply chain by focusing on the distribution and shipment of equipment, but this does not address the complete "chain." To address the emerging threat, the "supply chain" analysis must address all parts and components of a system early in the program, including firmware and software. It must also analyze the impact of people, purchase of substitute parts, and automated processes (e.g., software patching) on the supply chain processes.

Therefore, an accurate SCRM assessment includes an evaluation of the origin of the materiel, how it is distributed, and the government decision-making process in the selection of the product. The MITRE SE role is to ensure that the systems engineering process is applied to all components and parts of a system throughout their life cycle.

Life-cycle applicability. MITRE SEs should be prepared to apply SCRM at any point of a system's life; it is never too late nor too early in a system life to incorporate the SCRM process. SCRM is currently being applied to materiel supply during the logistic phases, but a more effective systems engineering process should include addressing SCRM as early in the program as possible.

The DoD CNCI SCRM pilot program produced an implementation guide that offers detailed suggestions on how and when SCRM should be integrated into the life cycle of a system. This guide was developed to assist SEs and explains how they can incorporate SCRM prior to design and throughout its life. A summary of some key steps identified in the guide that a MITRE SE should understand includes:

- Determine system criticality.
- Determine the supply chain threat.
- Select build versus buy.
- Select SCRM key practices and determine sufficiency.
- Understand the Risk Management Plan adopted by the government efforts they support.
- Understand the likelihood and the consequence of insufficient SCRM practices.

Systems engineering and SCRM. The core systems engineering process used to protect the supply chain is risk management (see Risk Management in the Acquisition Systems Engineering section of the SEG). Pilot programs selected by the DoD to help refine SCRM policy are using the Information and Communication Technology Supplier Risk Management Process. The concept of operations for the DoD Comprehensive National Cybersecurity Initiative Supply Chain Risk Management Pilot Program describes this process [5].

Though risk management establishes the core for an effective SCRM process, SEs should also understand the relationship of other systems engineering disciplines and processes to SCRM [6]. Standard program documentation addressing software engineering practices and procedures should include applicability to their SCRM process. Another process that supports the protection of the supply chain is configuration management. Through configuration control and management, SEs can ensure that the system's baseline is tightly controlled and any changes to it are in accordance with appropriate systems engineering rigor and review (see Configuration Management in the Acquisition Systems Engineering section of the SEG).

SEs should ensure that acquisition, sustainment, disposal, and other program documentation are properly updated to include SCRM. At a minimum, the following kinds of documents should incorporate the SCRM process and findings: Program Protection Plan, Systems Engineering Plans/Procedures, and Life Cycle Management Plans. In addition, SEs should work closely with contracts and legal staff to verify that SCRM is included as part of the acquisition documentation, source selection criteria, and contractual clauses. They should also ensure that the SCRM practices are included as part of the sustainment documentation, supplier selection criteria, purchasing clauses, incoming inspection, quality verification testing, acceptance for inventory, and disposal processes.

References and Resources

1. Mirsky, A., May 4, 2009, *Supply Chain Risk Management (SCRM)*.
2. National Security Presidential Directive 54/Homeland Security Presidential Directive 23, January 8, 2008, *National Cyber Security Initiative*, paragraph 45.
3. Extract from Public Law 110-417, October 14, 1008, "Duncan Hunter NDAA for Fiscal Year 2009."
4. Office of the Deputy Under Secretary of Defense for Logistics and Materiel Readiness, May 23, 2003, DoD 4140.1-R DoD Supply Chain Materiel Management Regulation.
5. SCRM PMO Globalization Task Force OASSA(NII)-CIO/ODASD(IIA), April 24, 2009, *Concept of Operations for the DoD Comprehensive National Cybersecurity Initiative Supply Chain Risk Management Pilot Program*, Ver. 2.0.
6. National Defense Industrial Association System Assurance Committee, October 2008, Engineering for System Assurance, Ver. 1.0.

Transformation Planning and Organizational Change

Definition: *Transformation planning is a process that develops a plan to modify an enterprise's business processes by modifying policies, procedures, and processes to move from an "as is" to a "to be" state. Change management is a process for gaining business intelligence to perform transformation planning by assessing an organization's people and cultures to determine how changes (e.g., to strategy, structure, process, or technology) will impact the enterprise.*

Keywords: *business transformation, complex systems, life-cycle development, organizational change management, organizational development, organizational strategy, organizational transformation, psychology, social sciences, stakeholder management, systems thinking, trust*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to assist in formulating the strategy and the plans for transforming a customer's engineering/technical organization, structure, and processes, including the MITRE support to that organization. MITRE SEs are expected to recommend interfaces and interactions with other organizations, lead change, collaborate, build consensus across the MITRE support and other stakeholders for the transformation, and to assist in communicating the changes. To execute these roles and meet these expectations, MITRE SEs are expected to understand the complex, open-systems nature of how organizations change, and the importance of developing the workforce transformation strategies as a critical, fundamental, and essential activity in framing a project plan. They must understand the social processes and other factors (e.g., leadership, culture, structure, strategy, competencies, and psychological contracts) that affect the successful transformation of a complex organizational system.

Context

The objective of organizational change management is to enable organization members and stakeholders to adapt to a sponsor's new vision, mission, and systems, as well as to identify sources of resistance to the changes and minimize resistance to them. Organizations are almost always in a state of change, whether the change is continuous or episodic. Change creates tension and strain in a sponsor's social system that the sponsor must adapt to so that it can evolve. Transformational planning and organizational change is the coordinated management of change activities affecting users, as imposed by new or altered business processes, policies, or procedures and related systems implemented by the sponsor. The objectives are to effectively transfer knowledge and skills that enable users to adopt the sponsor's new vision, mission, and systems and to identify and minimize sources of resistance to the sponsor's changes.

Best Practices and Lessons Learned

Implementation of a large-scale information technology (IT) transformation project affects the entire organization. In a technology-based transformation project, an organization often focuses solely on acquiring and installing the right hardware and software. But the people who are going to use the new technologies—and the processes that will guide their use—are even more important. As critical as the new technologies may be, they are only tools for people to use in carrying out the agency's work.

Integrate organizational change management principles into your program. As Figure 1 shows, the discipline of organizational change management (OCM) is intended to help move an organization's people, processes, and technology from the current "as is" state to a desired future "to be" state. To ensure effective, long-term, and sustainable results, there must be a transition during which the required changes are introduced, tested, understood, and accepted. People have to let go of existing behaviors and attitudes and move to new behaviors and attitudes that achieve

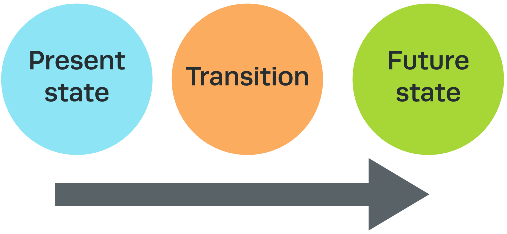


Figure 1. Organizational Transition Model

and sustain the desired business outcomes. That is why OCM is a critical component of any enterprise transformation program: It provides a systematic approach that supports both the organization and the individuals within it as they plan, accept, implement, and transition from the present state to the future state.

Studies have found that the lack of effective OCM in an IT modernization project leads to a higher percentage of failure. According to a 2005 Gartner survey on "The User's View of Why IT Projects Fail," the findings pinned the failure in 31 percent of the cases on an OCM deficiency. This demonstrates the importance of integrating OCM principles into every aspect of an IT modernization or business transformation program.

Commit to completing the change process. MITRE SEs need to assess change as a process and work in partnership with our sponsors to develop appraisals and recommendations to identify and resolve complex organizational issues. The change process shown in Figure 2 is designed to help assess where an organization is in the change process and to determine what it needs to do as it moves through the process.

By defining and completing a change process, an organization can better define and document the activities that must be managed during the transition phase. Moving through these stages helps ensure effective, long-term, and sustainable results. These stages unfold as an organization moves through the transition phase in which the required transformational changes are introduced, tested, understood, and accepted in a manner that enables individuals to let go of their existing behaviors and attitudes and develop any new skills needed to sustain desired business outcomes.

It is very common for organizations to lose focus or create new initiatives without ever completing the change process for a specific program or

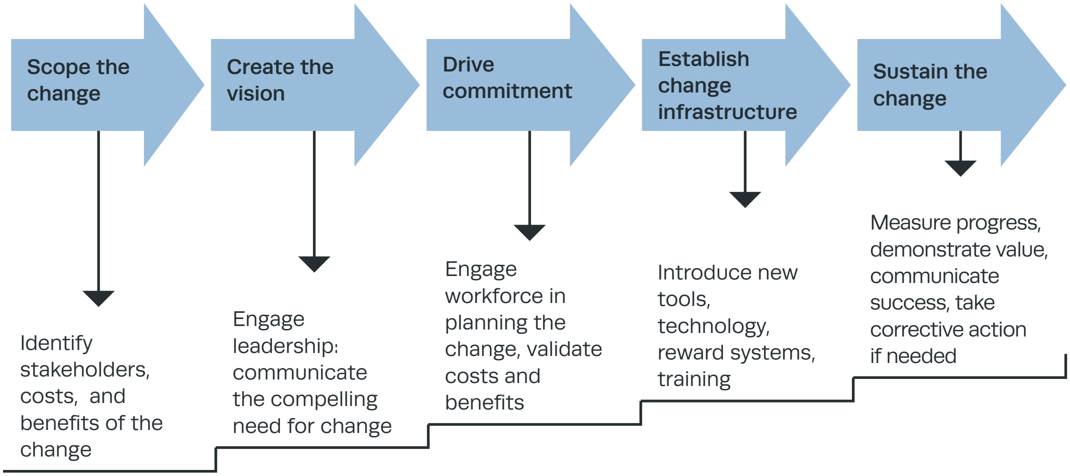

Scope the change
Create the vision
Drive commitment
Establish change infrastructure
Sustain the change
Identify stakeholders, costs, and benefits of the change
Engage leadership: communicate the compelling need for change
Engage workforce in planning the change, validate costs and benefits
Introduce new tools, technology, reward systems, training
Measure progress, demonstrate value, communicate success, take corrective action if needed

Figure 2. An Organizational Change Process

project. It is critical to the success of a transformation program for the organization to recognize this fact and be prepared to continue through the process and not lose focus as the organizational change initiative is implemented. Commitment to completing the change process is vital to a successful outcome. For more information, see the SEG article "Formulation of Organizational Transformation Strategies."

Establish a framework for change. In any enterprise transformation effort, a number of variables exist simultaneously and affect the acceptance of change by an organization. These variables range from Congressional mandates to the organization's culture and leadership to the attitude and behavior of the lowest ranking employee. At MITRE, social scientists use the Burke-Litwin Model of Organizational Performance and Change, or other approaches in line with the sponsor's environment and culture, to assess readiness and plan to implement change. The Burke-Litwin Model identifies critical transformational and transactional factors that may impact the successful adoption of the planned change. In most government transformation efforts, the external environment (such as Congressional mandates), strategy, leadership, and culture can be the most powerful drivers for creating organizational change. For more information, see the SEG article "Performing Organizational Assessments."

Align the transformation strategy with the organization's culture. Most organizations ultimately follow one of three approaches to transformation. The type of approach relates to the culture and type of organization (e.g., loosely coupled [relaxed bureaucratic organizational cultures] or tightly coupled [strong bureaucratic organizational cultures]):

- **Data-driven** change strategies emphasize reasoning as a tactic for bringing about a change in a social system. Experts, either internal or external to the sponsor, are contracted to analyze the system with the goal of making it more efficient (leveling costs vs. benefits). Systems science theories are employed to view the social system from a wide-angle perspective and to account for inputs, outputs, and transformation processes.

  The effectiveness of a sponsor's data-driven change strategy depends on (a) a well-researched analysis that the transformation is feasible, (b) a demonstration that illustrates how the transformation has been successful in similar situations, and (c) a clear description of the results of the transformation. People will adopt the transform when they understand the results of the transformation and the rationale behind it.
- **Participative** change strategies assume that change will occur if impacted units and individuals modify their perspective from old behavior patterns in favor of new behaviors and business/work practices. Participative change typically involves not just changes in rationales for action, but changes in the attitudes, values, skills, and percepts of the organization.

  To be successful, this change strategy depends on all impacted organizational units and individuals participating both in the change (including system design, development, and implementation of the change) and their change "re-education." The degree of success depends on the extent to which the organizational units, impacted users, and stakeholders are involved in the participative change transition plan.

- **Compliance-based** change strategies are based on the "leveraging" of power coming from the sponsor's position within the organization to implement the change. The sponsor assumes that the units or individuals will change because they are dependent on those with authority. Typically the change agent does not attempt to gain insight into possible resistance to the change and does not consult with impacted units or individuals. Change agents simply announce the change and specify what organizational units and impacted personnel must do to implement the change.

  The effectiveness of a sponsor's compliance-based change strategy depends on the discipline within the sponsor's chain of command, processes, and culture and the capability of directly and indirectly impacted stakeholders to impact sponsor executives. Research demonstrates that compliance-based strategies are the least effective.

Regardless of the extent of the organizational change, it is critical that organizational impact and risk assessments be performed to allow sponsor executives to identify the resources necessary to successfully implement the change effort and to determine the impact of the change on the organization. For more information, see the SEG article "Performing Organizational Assessments."

Distinguish leadership and stakeholders. MITRE SEs need to be cognizant of the distinction between sponsor executives, change agents/leaders, and stakeholders:

- **Sponsor executives:** Typically sponsor executives are the individuals within an organization who are accountable to the government. Sponsor executives may or may not be change leaders.
- **Change leaders:** Typically the change leader is the sponsor's executive or committee of executives assigned to manage and implement the prescribed change. Change leaders must be empowered to make sponsor business process change decisions, to formulate and transmit the vision for the change, and to resolve resistance issues and concerns.
- **Stakeholders:** Typically stakeholders are internal and external entities that may be directly (such as participants) or indirectly impacted by the change. A business unit's dependence on a technology application to meet critical mission requirements is an example of a directly impacted stakeholder. An external (public/private, civil, or federal) entity's dependence on a data interface without direct participation in the change is an example of an indirect stakeholder.

Both directly and indirectly impacted stakeholders can be sources of resistance to a sponsor's transformation plan. For more information, see the SEG articles "Stakeholder Assessment and Management" and "Effective Communication and Influence."

View resistance as a positive and integrative force. Resistance is a critical element of organizational change activities. Resistance may be a unifying organizational force that resolves the tension between conflicts that are occurring as the result of organizational change. Resistance feedback occurs in three dimensions:

- Cognitive resistance occurs as the unit or individual perceives how the change will affect its likelihood of voicing ideas about organizational change. Signals of cognitive resistance may include limited or no willingness to communicate about or participate in change activities (such as those involving planning, resources, or implementation).
- Emotional resistance occurs as the unit or individuals balance emotions during change. Emotions about change are entrenched in an organization's values, beliefs, and symbols of culture. Emotional histories hinder change. Signals of emotional resistance include a low emotional commitment to change leading to inertia or a high emotional commitment leading to chaos.
- Behavior resistance is an integration of cognitive and emotional resistance that is manifested by less visible and more covert actions toward the organizational change. Signals of behavioral resistance are the development of rumors and other informal or routine forms of resistance by units or individuals.

Resistance is often seen as a negative force during transformation projects. However, properly understood, it is a positive and integrative force to be leveraged. It is the catalyst for resolving the converging and diverging currents between change leaders and respondents and creates agreement within an organizational system. For more information, see the SEG articles "Performing Organizational Assessments" and "Effective Communication and Influence."

Create an organizational transition plan. As discussed earlier (Figure 1), successful support of individuals and organizations through a major transformation effort requires a transition from the current to the future state. Conducting an organizational assessment based on the Burke-Litwin Model provides strategic insights into the complexity of the impact of the change on the organization. Once the nature and the impact of the organizational transformation are understood, the transformation owner or champion will have the critical data needed to create an organizational transition plan.

Typically the content or focus of the transition plan comes from the insights gained by conducting a "gap" analysis between the current state of the organization (based on the Burke-Litwin assessment) and the future state (defined through the strategy and vision for the transformation program). The transition plan should define how the organization will close the transformational and transactional gaps that are bound to occur

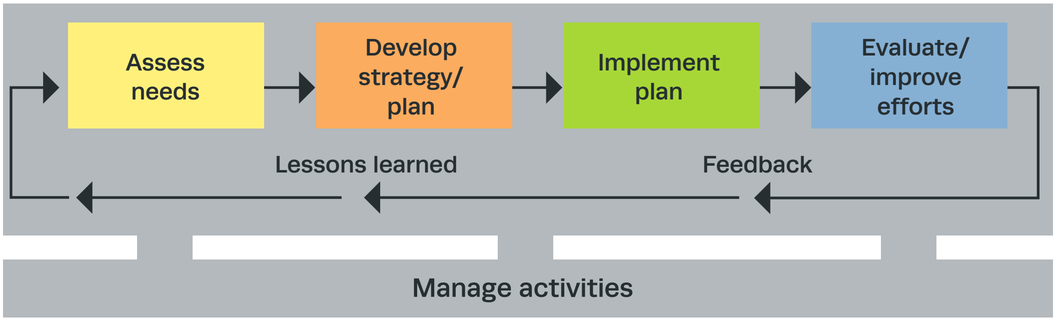


Figure 3. The Strategic Organizational Communications Process

during implementation of a transformation project. Change does not occur in a linear manner, but in a series of dynamic but predictable phases that require preparation and planning if they are to be navigated successfully. The transition plan provides the information and activities that allow the organization to manage this "nonlinearity" in a timely manner.

Large organizational change programs, which affect not only the headquarters location but also geographically dispersed sites, require site-level transition plans. These plans take into account the specific needs and requirements of each site. Most important, they will help "mobilize" the organizational change team at the site and engage the local workforce and leaders in planning for the upcoming transition.

Open and frequent communication is essential to effective change management. A key component of the transition plan should address the strategic communications (Figure 3) required to support the implementation of the transformation. When impacted individuals receive the information (directly and indirectly) they need about the benefits and impact of the change, they will more readily accept and support it. The approach to communication planning needs to be integrated, multi-layered, and iterative. For more information, see the SEG article "Effective Communication and Influence."

References and Resources

Burke, W., 2008, *Organizational Change: Theory and Practice.* Sage Publications, 2nd Ed.

Burke, W., and G. Litwin, 1992. "A Causal Model of Organizational Performance and Change," *Journal of Management,* Vol. 18, No. 3.

Flint, D., 2005, "The User's View of Why IT Projects Fail," Gartner Report.

Kotter, John P., 1998, "Winning at Change," *Leader to Leader,* 10 (Fall 1998), 27–33.

Lawson, E., and C. Price, 2003, "The Psychology of Change Management," *McKinsey Quarterly.*

Kelman, S., 2005, *Unleashing Change: A Study of Organizational Renewal in Government*, The Brookings Institute.

Mergers and Transformations: Lessons Learned from DHS and Other Federal Agencies, November 2002, GAO-03-293SP.

Ostroff, F., May 2006, “Change Management in Government,” *Harvard Business Review.*

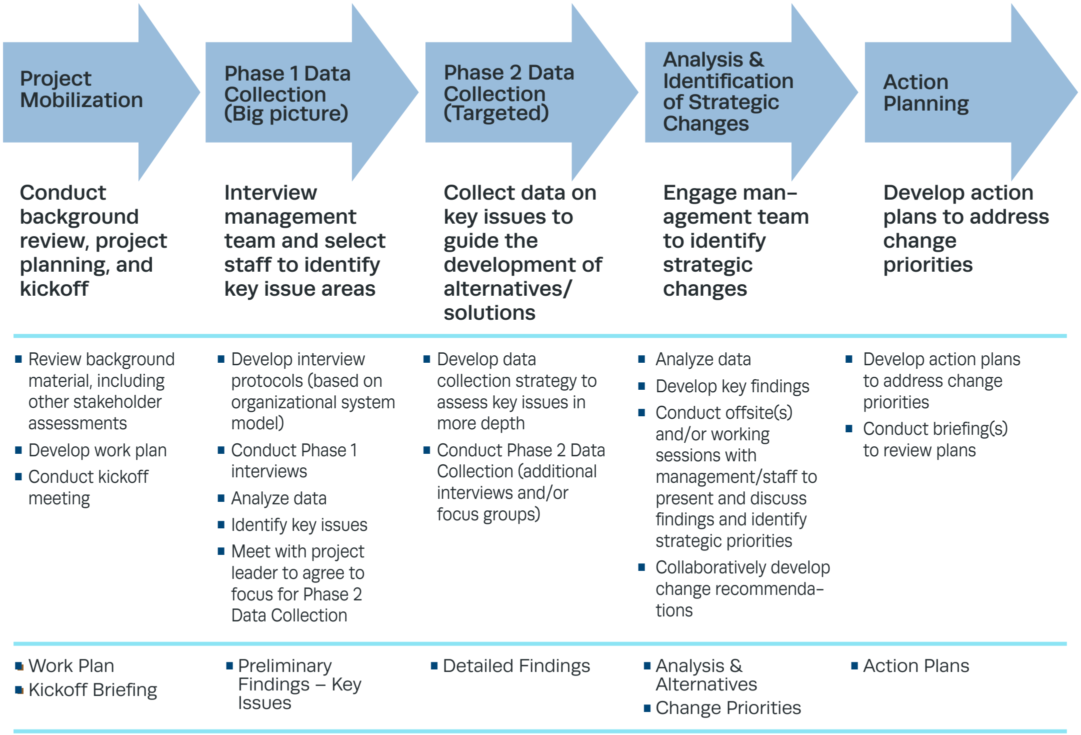
Definition: *Organizational assessments follow a systems science approach to analyze a proposed transformation, determine the impacts of the transformation on the organization, assess the preparedness of the organizational entities to adopt the transformation, and assess the "people and organizational" risks associated with the transformation.*

Keywords: *business intelligence, direct and indirect stakeholders, organizational impacts, organizational risk, strategic alignment*

TRANSFORMATION PLANNING AND ORGANIZATIONAL CHANGE

Performing Organizational Assessments

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be cognizant of the behavioral complexities of transformation on organizations, the necessity to analyze the organization's capability, to understand organizational drivers that will impact the transformation, and how to align the organization to successfully adopt the change. MITRE uses organizational assessments to provide sponsor executives and managers the business intelligence to successfully lead the transformation. MITRE SEs are expected to develop and recommend organizational strategies that will facilitate the successful adoption of the change, and to monitor and evaluate sponsor organizational change management (OCM) efforts recommending changes as warranted.



Project Mobilization	Phase 1 Data Collection (Big picture)	Phase 2 Data Collection (Targeted)	Analysis & Identification of Strategic Changes	Action Planning
Conduct background review, project planning, and kickoff	Interview management team and select staff to identify key issue areas	Collect data on key issues to guide the development of alternatives/ solutions	Engage management team to identify strategic changes	Develop action plans to address change priorities
▪ Review background material, including other stakeholder assessments ▪ Develop work plan ▪ Conduct kickoff meeting	▪ Develop interview protocols (based on organizational system model) ▪ Conduct Phase 1 interviews ▪ Analyze data ▪ Identify key issues ▪ Meet with project leader to agree to focus for Phase 2 Data Collection	▪ Develop data collection strategy to assess key issues in more depth ▪ Conduct Phase 2 Data Collection (additional interviews and/or focus groups)	▪ Analyze data ▪ Develop key findings ▪ Conduct offsite(s) and/or working sessions with management/staff to present and discuss findings and identify strategic priorities ▪ Collaboratively develop change recommendations	▪ Develop action plans to address change priorities ▪ Conduct briefing(s) to review plans
▪ Work Plan ▪ Kickoff Briefing	▪ Preliminary Findings – Key Issues	▪ Detailed Findings	▪ Analysis & Alternatives ▪ Change Priorities	▪ Action Plans

Figure 1. MITRE's Organizational Assessment Approach

Background

Organizational assessments follow a system science approach to assess the dynamics at work in the sponsor's organization. The approach is to collect data and analyze factors that impact organizational performance to identify areas of strength as well as opportunity. There are a number of excellent models for understanding and analyzing data during an organizational change assessment, including the Burke-Litwin Model of Organizational Performance and Change shown in Figure 2. This model has a number of interdependent factors, both external and internal, that exist simultaneously and affect the performance of an organization. These interdependent variables range from external budget pressures, to the organization's culture and leadership, to the skills and behavior of the lowest level employee. The Burke-Litwin model provides a framework to effectively analyze, interpret, develop recommendations, communicate, and manage change within an organization.

Best Practices and Lessons Learned

MITRE's organizational assessment approach. One organizational assessment approach that MITRE uses is shown in Figure 1. The assessment is a repeatable process that applies social behavioral best practices developed and proven effective in the public and private sectors. This process is designed to help leaders assess where their organization is in the change process, identify organizational gaps, transformation risks/issues, and to determine what they need to do as they move through the process.

The following five-step approach is suggested:

1. **Project Mobilization—**During the start-up phase of the project, review background material, conduct project planning, and conduct initial meetings with the client to gain insights and discuss the project approach.
   Deliverables: Work plan and kickoff briefing
2. **Phase 1 Data Collection (Big Picture)—** The first phase of data collection will provide a holistic, big picture assessment of the organization. Working with an organization's leadership, identify

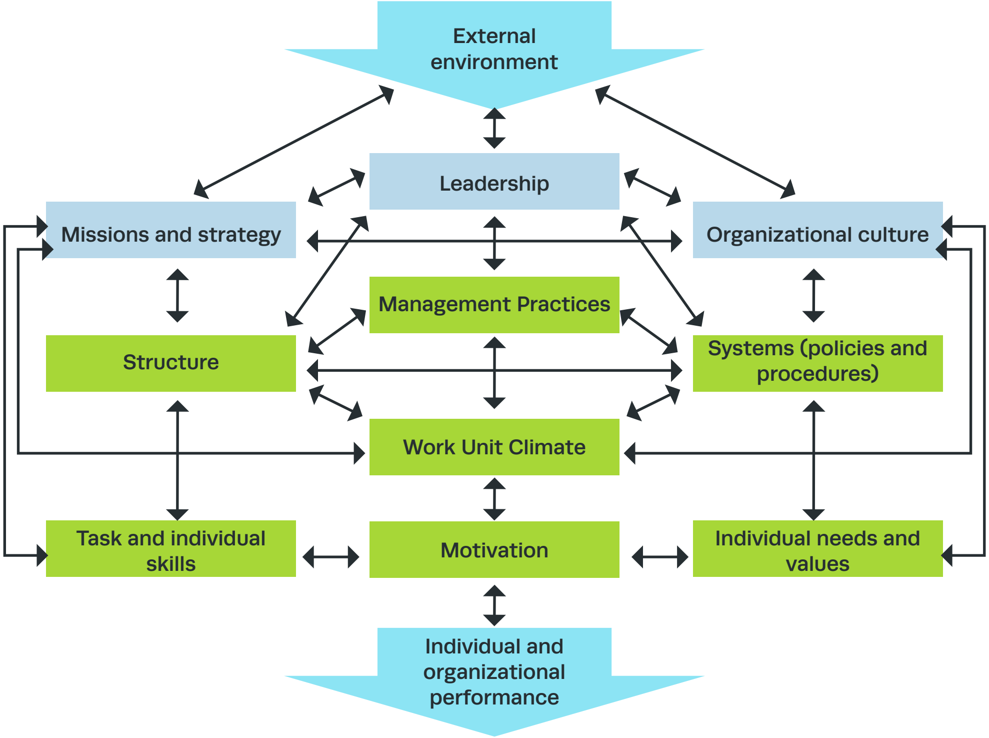


Figure 2. Burke-Litwin Model of Organizational Performance and Change

key stakeholders to interview. Develop interview protocols based on an organizational systems model and investigate such areas as External Environment, Mission and Strategy, Leadership, Organizational Culture, Organizational Structure, Management Practices/Processes, and any specific areas of interest and need to the organization. Collect and analyze data, and identify key issue areas.
Deliverable: Preliminary findings—key issues

3. **Phase 2 Data Collection (Targeted)—** After discussion and agreement with organizational leaders, conduct a second phase of data collection to gather more in-depth understanding around key issue areas to guide the development of alternatives and solutions.
   Deliverable: Detailed findings
4. **Analysis and Identification of Strategic Changes—**After analyzing the more detailed data, engage the organization's management team in a process to identify strategic changes (through offsites and/or working sessions).
   Deliverable: Analysis and alternatives
5. **Action Planning—**If desired, collaborate with an organization's management team to develop action plans to address change priorities.
   Deliverable: Action plans

Note: All organizational assessments require sponsor participation and direction on the goals and objectives of the transformation prior to performing the analysis of workforce.

Burke-Litwin Model of organizational performance and change (Figure 2). A system science model that describes the linkages among the key factors that affect performance, and determines how change occurs in an organization. SEs use this model system to obtain data on what organizational factors to change and why. Higher level factors (blue boxes) have greater weight in effecting organizational change; a change in any variable ultimately affects every other variable. Table 1 provides key sample questions SEs should ask regarding the 12 variables, or dimensions, of the Burke-Litwin Model.

Table 1. Dimensions of Burke-Litwin Model

Dimensions of Model	Key Questions
1. External Environment	What are the key external drivers? How are these likely to impact on the organization? Does the organization recognize these?
2. Mission and Strategy	What does top management see as the organization's mission and strategy? Is there a clear vision and mission statement? What are employees' perceptions of these?

3. Leadership	Who provides overall direction for the organization? Who are the role models? What is the style of leadership? What are the perspectives of employees?
4. Organizational Culture	What are the overt and covert rules, values, customs, and principles that guide organizational behavior?
5. Structure	How are functions and people arranged in specific areas and levels of responsibility? What are the key decision-making, communication, and control relationships?
6. Systems	What are the organization's policies and procedures, including systems for reward and performance appraisal, management information, human resources, and resource planning?
7. Management Practices	How do managers use human and material resources to carry out the organization's strategy? What is their style of management, and how do they relate to subordinates?
8. Work Unit Climate	What are the collective impressions, expectations, and feelings of staff? What is the nature of relationship with work unit colleagues and those in other work units?
9. Task and Individual Skills	What are the task requirements and individual skills/abilities/knowledge needed for task effectiveness? How appropriate is the organization's "job-person" match?
10. Individual Needs and Values	What do staff members value in their work? What are the psychological factors that would enrich their jobs and increase job satisfaction?
11. Motivation	Do staff feel motivated to take the action necessary to achieve the organization's strategy? Of factors 1 through 10, which seem to be impacting motivation the most?
12. Individual and Organizational Performance	What is the level of performance in terms of productivity, customer satisfaction, quality, and so on? Which factors are critical for motivation and therefore performance?

Organizational Assessment Products

Primary outputs from the organizational assessments include:

Organizational Impact Assessment (OIA). Provides information on the status of the organizational entities and personnel to adopt the transformation. The OIA will identify direct and indirect impacts on the workforce, direct and indirect stakeholders and how the transformation will impact the accomplishment of the sponsor's mission.

Organizational Risk Assessment (ORA). Provides sponsor executives with business intelligence on the type and severity of transformation risks and issues and potential mitigation solutions. The ORA may be integrated into one overall organizational impact assessment. *Note: The organizational change strategy output from the OIA and ORA provide sponsor executives with the business intelligence to develop the organizational change management (OCM) direction.*

[optional] Deliverable-Workforce Transformation Strategy and Plan. Explains the transformation plan ensuring integration with the sponsor's technical and deployment teams, integrates organization preparation, communication, and training activities into one transformation plan, and explains how the transformation program management team will manage daily OCM activities and briefings.

SEs also need to be cognizant that a system science approach includes communications planning and outreach strategies to initiate and sustain communications to affected organizational entities and key transformation participants (e.g., internal and external stakeholders). Communications planning requires the development of near-term communications and subsequent implementation of the plans. For more information, see the SEG article "Effective Communication and Influence."

References and Resources

Burke, W., 2008. *Organizational Change: Theory and Practice,* 2nd Ed., Sage Publications.

Burke, W., and G. Litwin, 1992, "A Casual Model of Organizational Performance and Change," *Journal of Management,* Vol. 18, No. 3.

Flint, D., 2005, "The User's View of Why IT Projects Fail," *Gartner Report*.

Kelman, S., 2005, *Unleashing Change: A Study of Organizational Renewal in Government*, The Brookings Institute.

Kotter, J. P., 1998, "Winning at Change," *Leader to Leader,* 10 (Fall 1998), 27–33.

Lawson E. and C. Price, 2003, "The Psychology of Change Management," *McKinsey Quarterly.*

Mergers and Transformations: Lessons Learned from DHS and Other Federal Agencies, November 2002, GAO-03-293SP.

Ostroff, F., May 2006, "Change Management in Government," *Harvard Business Review.*

TRANSFORMATION PLANNING AND ORGANIZATIONAL CHANGE

Formulation of Organizational Transformation Strategies

Definition: *The formulation of an organizational transformation strategy documents and institutionalizes the sponsor's commitment and the strategic approach to the transformation. The formulation of the transformation strategy provides the foundation on which the sponsor's change agents will assist affected organizational units and users to align and adapt to the transformation.*

Keywords: *change management, organizational alignment, organizational change*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be cognizant of the complexities of organizational transformation. They are expected to be able to formulate an organizational transformation strategy that considers the human dimension of a technology modernization effort.

Background: Why Projects Fail

According to a 2005 Gartner survey of failed information technology projects, in 31 percent of cases, failure was due to a deficiency in organizational change management. In addition, in 44 percent of failed projects, organizational change problems were identified as part of the reason for project failure. Essentially, the degree to which an organization's management is able to manage change, develop consensus, and sustain commitment will determine the success or failure of any large enterprise modernization effort.

The formulation of an organizational transformation management strategy is a critical component of any modernization program and involves a systematic approach that supports both the organization and the individuals in it to plan for the change and then accept, implement, and benefit from the change.

As government agencies expand and improve their services, they may undergo a fundamental transformation of mission, strategy, operations, and technology. If managed effectively, these changes can increase the quality of government services and reduce taxpayer costs. For most large government modernization programs, an organization's predominant focus is often on the technology. If a program's success depended solely on installing the right hardware and software, however, many more modernization programs would be successful. It is the people who are going to use the new technologies who add an unpredictable, complex dimension. The following best practices suggest approaches for the development of organizational transformation strategies.

Best Practices and Lessons Learned

Framework for Formulating Organizational Transformation: Burke–Litwin Model of Organizational Performance and Change. During an enterprise modernization effort, a number of variables exist simultaneously that affect the acceptance of change within an organization. The Burke–Litwin model (B–L) is a framework to assess the scope and complexity of these variables within an organization. As a model of organizational change and performance, B–L provides a link between an assessment of the wider institutional context and the nature and process of change within an organization. The B–L model identifies these key factors to consider during organizational change:

- The external environment is the most powerful driver of organizational change.
- The changes that occur in the external environment lead to "transformational" factors within an organization—mission and strategy, organizational culture, and leadership.
- The changes in transformational factors lead to changes in the "transactional" factors within an organization—structure,

systems, management practices, and organizational climate.

- Together, changes in transformational and transactional factors affect motivation, which in turn affects individual and organizational performance.

For an enterprise modernization effort to be effective and sustainable, changes in transformational and transactional factors need to be integrated and consistent. Experience and practice suggest that the variables highlighted in the model and the relationships between them provide a useful tool for communicating not only how organizations perform, but how to effectively implement change. For more information, see the SEG article "Performing Organizational Assessments."

Elements of Organizational Transformation Strategy. Organizational transformation relies on five key elements— leadership, communications and stakeholder engagement, enterprise organizational alignment, education and training, and site-level workforce transition—that provide an overall framework for change. Each of these elements is considered a "work stream" in the transformation strategy, which are addressed in later sections of this article. The fifth work stream, site-level workforce transition, incorporates the first four elements and applies them at the level of the affected site or geographic region to prepare and manage users in the field through the implementation effort.

Figure 1 shows the assessment approach used to formulate the organizational transformation strategy. This approach has been found to enhance the formulation of organizational transformation strategies. When used in concert, the elements create a powerful, mutually reinforcing field for the support of organizational change and improve the chances that the transformation will meet its objectives. These elements of change leverage the resources within the sponsor's organization to reduce the risks and address the challenges outlined above. For more information, see the SEG article "Performing Organizational Assessments."

MITRE SEs must understand that the development of organizational transformation strategies involves the following assessments:

- **Leadership:** Assess the sponsor's leadership. Mobilizing leaders is critical to spearheading a successful effort. Leaders play a vital role throughout the life cycle in promoting the initiative, ensuring resources are available and able to support the effort, and resolving critical implementation issues as they arise. Leaders must be aware of outcomes across the organization and be able to make decisions accordingly.
- **Communications and Stakeholder Engagement:** Identify key stakeholders (those who will be impacted), determine how best to communicate with them, and keep them involved. Effective communications allow for two-way dialogue, so issues can be understood, and changes can be made appropriately. Assess access to stakeholder information. Access to stakeholder information is critical to the training team, which must determine which groups need to be trained and how. For more information, see the SEG article "Effective Communication and Influence."

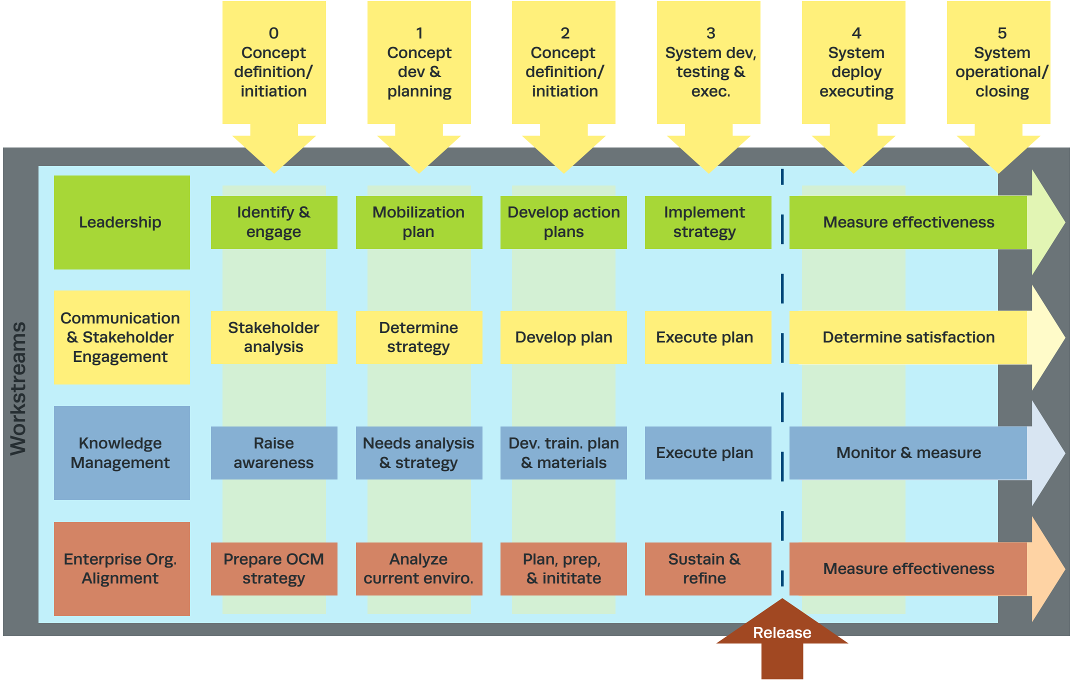


Figure 1. Organizational Change Management Framework

- **Knowledge Management:** Assess directly and indirectly affected users to determine if they are prepared to adopt the transformation. While training is delivered just prior to "going live," education needs to occur much sooner. End users must understand what is changing and why, before they are trained on "how." This assessment is tightly linked with leadership and communication assessments. For more information, see the SEG articles "Performing Organizational Assessments," "Stakeholder Assessment and Management," and "Effective Communication and Influence."
- **Enterprise Organizational Alignment:** Assess the sponsor's organization to determine how the transformation will specifically affect the organization and any external stakeholders. The transformation may be creating new organizational units or user roles to be filled by current employees. The Burke-Litwin analysis will identify current organizational gaps. Understanding the gap between present and future roles and responsibilities is critical to prepare the organization to successfully adopt the change. For more information, see the SEG article "Performing Organizational Assessments."

- **Site-Level Workforce Transition:** The relationship between headquarters and field offices adds complexity to the organizational assessment. Systems engineers must be cognizant of the need to assess field offices as part of the overall organizational assessment. The success of organizational changes to each site will depend on the degree of involvement by its local team. Each site likely has its own processes, issues, constraints, and numbers of people affected. Therefore, they must each be accountable for developing a transition plan that is tailored to meet their needs. For more information, see the SEG articles "Stakeholder Assessment and Management" and "Performing Organizational Assessments."

These work stream assessments create a comprehensive blueprint for the formulation of an organizational transformation strategy to increase the likelihood of transformation success.

References and Resources

Baba, M., March 6, 2005, *The Defense Logistics Enterprise: Transforming Organizations in the Information Era*, Prepared for Enterprise Integration Group.

Burke, W., and Litwin, G., 1992, "A Causal of Organizational Performance and Change, *Journal of Management*, 18(3), pp. 523–545.

Gartner, 2005, "The User's View of Why IT Projects Fail," *Research Notes,* 2, 4.

TRANSFORMATION PLANNING AND ORGANIZATIONAL CHANGE

Stakeholder Assessment and Management

Definition: *Stakeholder management is the process of identifying stakeholder groups, the interests they represent, the amount of power they possess, and determining if they represent inhibiting or supporting factors toward the transformation. The objective of stakeholder planning and management is determining who the stakeholders are and how they should be dealt with [1].*

Keywords: *communications, interfaces, monitoring, power, risk development, stakeholders*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) must understand the importance of identifying both directly and indirectly impacted stakeholders in transformation planning and organizational change. Systems engineers must assess the impact of transformation on people and the organization to identify all stakeholders, identify transformation risks and issues, rank the risks associated with the transformation, and recommend mitigation strategies to sponsor executives. MITRE SEs should work closely with the sponsor's communications team to promote transformation awareness, understanding, and acceptance across key stakeholder groups.

Gaining the support of key stakeholders is critical to creating successful organizational change efforts [2].

Best Practices and Lessons Learned

Build trust with stakeholders. The Organizational Change Management Practice (OCM) in MITRE's Center for Connected Government (CCG) has conducted several stakeholder analyses on behalf of our sponsors. Lessons learned show that the following characteristics are common among effective stakeholder relationships:

- A deep level of trust with the change sponsor and the stakeholder groups affected by the change initiative
- Effective communication with the stakeholders allowing them to gain a new understanding of the benefits and costs of the change
- Close change sponsor and change agents' relationships that allow them to become personally engaged in and committed to initiatives based on the findings and recommendations.

Identify stakeholders. MITRE SEs must ensure that representatives from all key stakeholders are included in the organizational impact assessments and that the assessment information collected is representative of the affected population. It is important to take into account geographic distribution of stakeholder groups to obtain the full range of perspectives.

During transformation planning, identify all key stakeholder groups, including:

- Decision makers involved in the decisions regarding the change
- Change sponsors and agents responsible for executing the change
- Employees [and contractors] directly impacted by the change
- "Customers" of the change agents affected by the change
- Representatives of all groups in headquarters and in field offices across the country (as appropriate).

Starting from a list of individuals and organizations provided by the sponsor, the stakeholder assessment team should ask interviewees and others within the sponsor's organization to provide suggestions for additional names and organizations to be included. The stakeholder assessment should seek out and integrate input from supporters, skeptics, and rivals of the transformation initiative.

Collect and analyze data. MITRE SEs must build relationships with the stakeholders throughout the transformation process [3]. They should employ a combination of one-on-one interviews, focus groups, and surveys to rapidly establish rapport and create an environment that contributes to the stakeholders being open and honest while describing challenging situations.

Rather than just fire off one question after the next, it is important to engage stakeholders in dialogue and exhibit interest in their opinions and perspectives. Ask follow-up questions to solicit specific examples and understand how stakeholders developed their opinions and perceptions. The interview protocol should include open-ended and Likert-scaled questions. Likert scales are a type of survey response format where survey respondents are asked to indicate their level of agreement/interest on a continuum (e.g., from strongly agree to strongly disagree or a numerical scale). This method provides a way to assign a quantitative value to qualitative information. Although there is a certain amount of variance inherent in Likert responses, these questions help bring a quantitative measure to enhancing understanding of stakeholders. In addition to asking probing questions on a variety of topics, solicit suggestions for addressing concerns.

Maintain detailed notes and analyze the information for key themes. Analyze the data to develop stakeholder maps (Figure 1) to graphically display the relative influence and support that stakeholder groups have for the transformation. Overlay the quantitative (Likert data) to identify similarities and differences across the stakeholder networks.

Present findings. Best practices for developing and presenting stakeholder findings include:

- Include history and context up front: This approach establishes a common understanding of the sponsor's challenging environment.
- Provide "good news": Sharing success stories acknowledges what is going well and contributes to a balanced message. Change sponsors are not trying to make life difficult for their stakeholders; they are focused on achieving specific business objectives that may have been initiated by Congress or other external factors.
- Present key themes and ground findings with specific examples: Identify overarching themes based on data analysis and

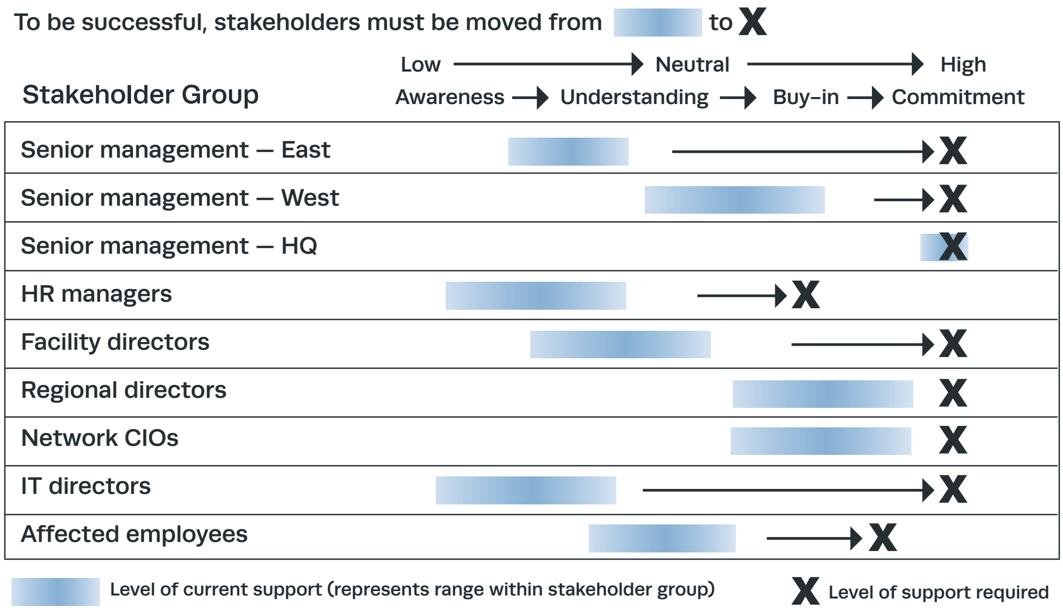


Figure 1. Stakeholder Findings

include supporting evidence (including Likert data), examples, and quotes [4, 5].

- Highlight differences across stakeholder groups (see Figure 1): Avoid making generalizations and note when opinions are consistent or divergent across stakeholder groups.
- Provide general information about the change process: Sponsors are more open to receiving challenging feedback when they understand that their situation is not unique. Be careful not to overwhelm sponsors with too much theory.
- Share recommendations and/or next steps: Findings are only useful in the context of how they can address issues and concerns. Sometimes, sponsors need to ponder the findings before engaging in recommendations. In that case, identifying next steps is helpful.
- Present the findings by telling stories using the key themes and supporting data: This approach will enable the data to come alive for different stakeholder groups. Identify a few common themes when reporting findings to sponsors.

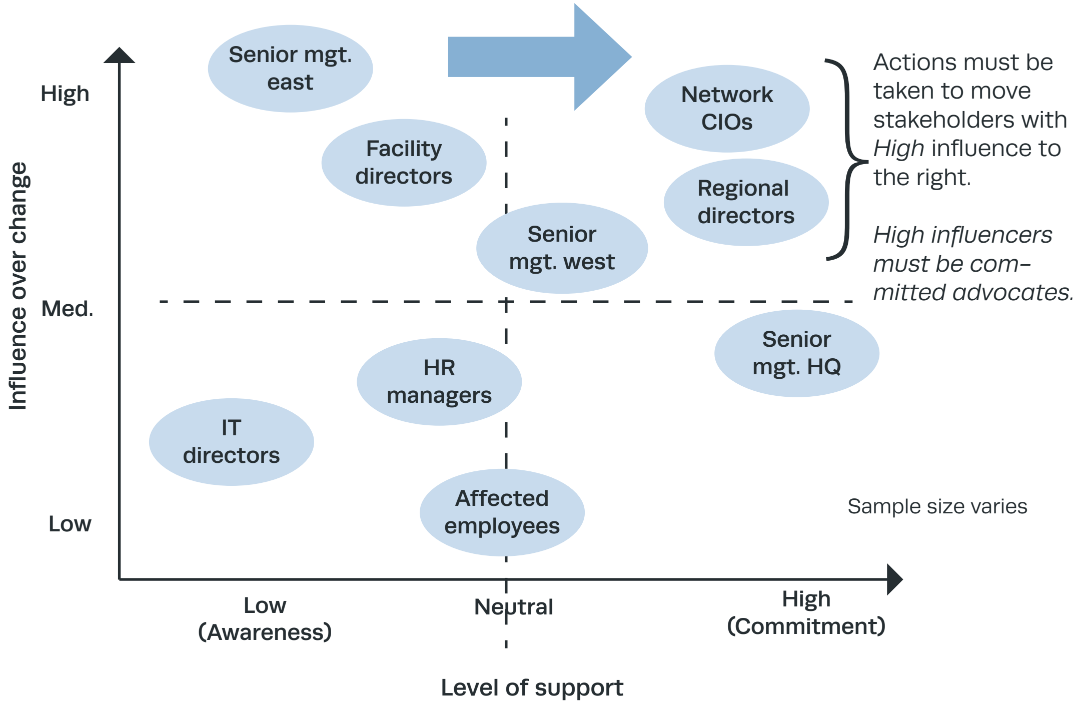


Figure 2. Current vs. Required Level of Support

Note: Before the stakeholder assessment, the executive change sponsors may perceive the employees as resistant. As a result of the stakeholder assessment findings, they may realize that the perceived resistance was really a lack of understanding about the rationale for change and how to operationalize it.

When MITRE SEs are asked to present the results from the stakeholder analysis to stakeholder groups, MITRE's approach should help build credibility. Stakeholders appreciate having their perspectives accurately presented to senior management and the value transparency that results from seeing the findings gathered across the stakeholder groups.

Summary

Stakeholders are a critical asset that may have a significant impact on transformation initiatives. Stakeholder analysis is an effective method for enabling different stakeholder groups to understand each other's perspectives and concerns. Establishing trust and credibility throughout the process—from planning and gathering to analyzing and presenting data—is critical to ensuring that the findings are valued and acted on.

References and Resources

1. Gardner, J., R. Rachlin, and H. Sweeny, 1986, *Handbook of Strategic Planning*, John Wiley.
2. Ostroff, F., 2006, "Change Management in Government," Harvard Business Review, 84(5), 141–147.
3. Maister, D., C. Green, and R. Galford, 2000, *The Trusted Advisor,* Simon & Schuster, New York, NY.
4. Kassinis, G., and N. Vafeas, 2006, "Stakeholder Pressures and Environmental Performance," *Academy of Management Journal*, 49(1), 145–159.
5. Mitchell, R., B. Agle, and D. Wood, 1997, "Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts," *Academy of Management Review*, 22, 853–886.

Definition: *Communication is a two-way process in which there is an exchange of thoughts, opinions, or information by speech, writing, or symbols toward a mutually accepted goal or outcome [1].*

Keywords: *behavior, behavior change, communication, elaboration, elaboration likelihood model (ELM), influence, message, persuasion, processing, social*

TRANSFORMATION PLANNING AND ORGANIZATIONAL CHANGE

Effective Communication and Influence

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) need to understand that communication is vital for transformation success. They are expected to assist in developing communication strategies as part of organizational transformations. SEs are expected to be able to communicate and interpret the "big picture" across multiple disciplines, teams, and environments. They translate the visions of leaders into the engineering work performed by technical staff, and communicate information broadly and accurately to achieve project success. In addition to being able to express ideas in a clear and compelling manner, they must adjust their language and communication vehicles to capture the attention of diverse audiences. Within their project teams, MITRE SEs are expected to help establish an environment of trust by communicating openly and behaving consistently in words and actions. In addition, they are expected to communicate effectively to persuade or influence others outside their formal authority to accept a point of view, adopt a specific agenda, or take a course of action that is in the best interests of the sponsor and the wider stakeholder community. They are expected to be proficient in analyzing audiences, organizing ideas effectively, choosing appropriate media, and knowing how to promote ideas to a wide range of audiences. Proper use of a systems science approach to communication will help build good relationships with team members, sponsors, and other key stakeholders, to increase the likelihood of project success.

Elements of Communication

In supporting program management, a key *"purpose of effective communication is sustaining the on-going work with maximum efficiency* [1]." This goes beyond just sending progress reports and providing periodic briefings. It also includes communicating high-quality standards for the project team, clearly communicating the financial impact of the work, and defining the outcome and benefits that stakeholders can expect [1, p. 1]. It involves facilitating broad participation in the decision-making process, thereby increasing support and commitment to the project. SEs who carry out these activities effectively may better accomplish the project objectives and lead to a more collaborative relationship with the customer [1, p. 2].

Strategic Communication Planning in Government Agencies

As government agencies expand and improve their services, they often undergo a fundamental transformation of mission, strategy, operations, and technology. If managed effectively, these changes can increase the quality of government services and reduce taxpayer costs. Often there is strong resistance to change within an organization because it requires that people change not only the way they work, but their attitudes and beliefs. This is challenging, but such transformation is essential if government agencies are to provide the high quality of service expected by citizens and mandated by legislators.

For most large government modernization programs, an organization's predominant focus is often on the enabling technology—the definition, acquisition, and implementation of information technology systems. If a program's success depended solely on installing the right hardware and software, however, many more modernization programs would be successful. It is the people who are going to use the new technologies who add an unpredictable, complex dimension.

The Role of Communication in Transformation Projects

People don't like change—they fear the unknown, fear losing control, worry that their jobs may change or go away. Affected individuals may oppose a transformation program by refusing to implement the necessary changes in their daily operations or by speaking against it and influencing others to oppose it. If the benefits of the transformation are not communicated broadly and consistently, departments and business units may refuse to support it. They lose sight of the overall mission of the agency, compete with each other for funds and resources, and refuse to see the value of collaborating with other units.

Open and frequent communication is an essential factor in successful transformation. Give people the information they need about the benefits and impact of the transformation, and they will more readily accept and support the effort. Leaders of transformation programs need a strategy that incorporates the communication needs of key stakeholders, the resources and channels required to reach these audiences, and the processes that support an understanding of the goals and benefits of the transformation program.

Best Practices and Lessons Learned

Figure 1 shows an approach to developing effective communications and influencing sponsor interactions. It incorporates industry best practices as well as MITRE lessons learned and proven methodologies in supporting government agencies and private sector organizations. It depicts a four-step systems approach to developing a communications strategy, building an action plan, executing the action plan, measuring feedback to assess the effectiveness of communication activities, and integrating feedback into revisions of the communication activities to improve their effectiveness.

Developing the communication strategy. The requirement for a communication strategy may be triggered by a variety of events: a new administration or agency leadership, a significant change in an agency's mission, or a legislative mandate that requires significant reorganization or modernization of an agency's operations or systems. In each case, the failure to consider the "human dimension" in a transformation leads to a higher percentage of failure.

The application of a systems science approach is effective in helping agencies understand the human dimensions of their transformation and in understanding how to effectively integrate communications into the transformation program. This approach requires that systems engineers listen carefully to the sponsor's needs and concerns, and then collaborate with them to develop and validate an effective communication strategy by:

- Assessing and analyzing both the communication needs of the agency's key audiences (internal and external stakeholders) and their concerns regarding the proposed transformation effort.

- Developing clarity about the goals and objectives that the communication effort is intended to accomplish.
- Establishing governance, with clear roles and responsibilities for those involved in the communication effort.
- Conducting an audit to determine existing internal and external communication resources and channels, and identify opportunities for new resources and channels.
- Identifying a measurement process and feedback mechanisms to ensure that the strategy is achieving its goals.

Developing the communication plan. After the communication strategy has been validated and accepted, the next step is to develop and implement a communication action plan to deliver key messages to each audience—in the language and through the channels that are most effective for each group. Development of the plan could include the following steps:

1. Determine the activities needed.
2. Develop key messages targeted to specific audiences and establish the process through which these messages will be reviewed and approved; this concurrence process may vary according to the audience (e.g., internal vs. external, legislative vs. media, etc.).
3. Identify the resources and/or communication channels that may be required and that would be most effective for each audience (e.g., electronic vs. print vs. in-person meetings; mailings and phone calls; website and social media such as Facebook).
4. Establish a detailed timeline for delivery of the messages, with sequenced delivery and multiple impressions for greatest effect.

Executing communication activities. The execution of communication activities begins when triggering events occur (i.e., program milestones that will affect users/stakeholders, system deployment, or significant unplanned events). The execution of communication activities should include the following steps:

- Review planned communication activities with the sponsor and revise the plan to address the specific triggering event.
- Obtain the needed media/channel resources.
- Draft the material.
- Perform internal edits and reviews.
- Execute concurrence process.
- Assign delivery dates.
- Execute communication activity.

Measuring effectiveness. Measuring the effectiveness of communication activities is a critical component in the development of an effective communication strategy. The measurement process should include the following steps:

1. Define measures of effectiveness and a measurement process to assess the results of various messages, channels, and delivery schedules.

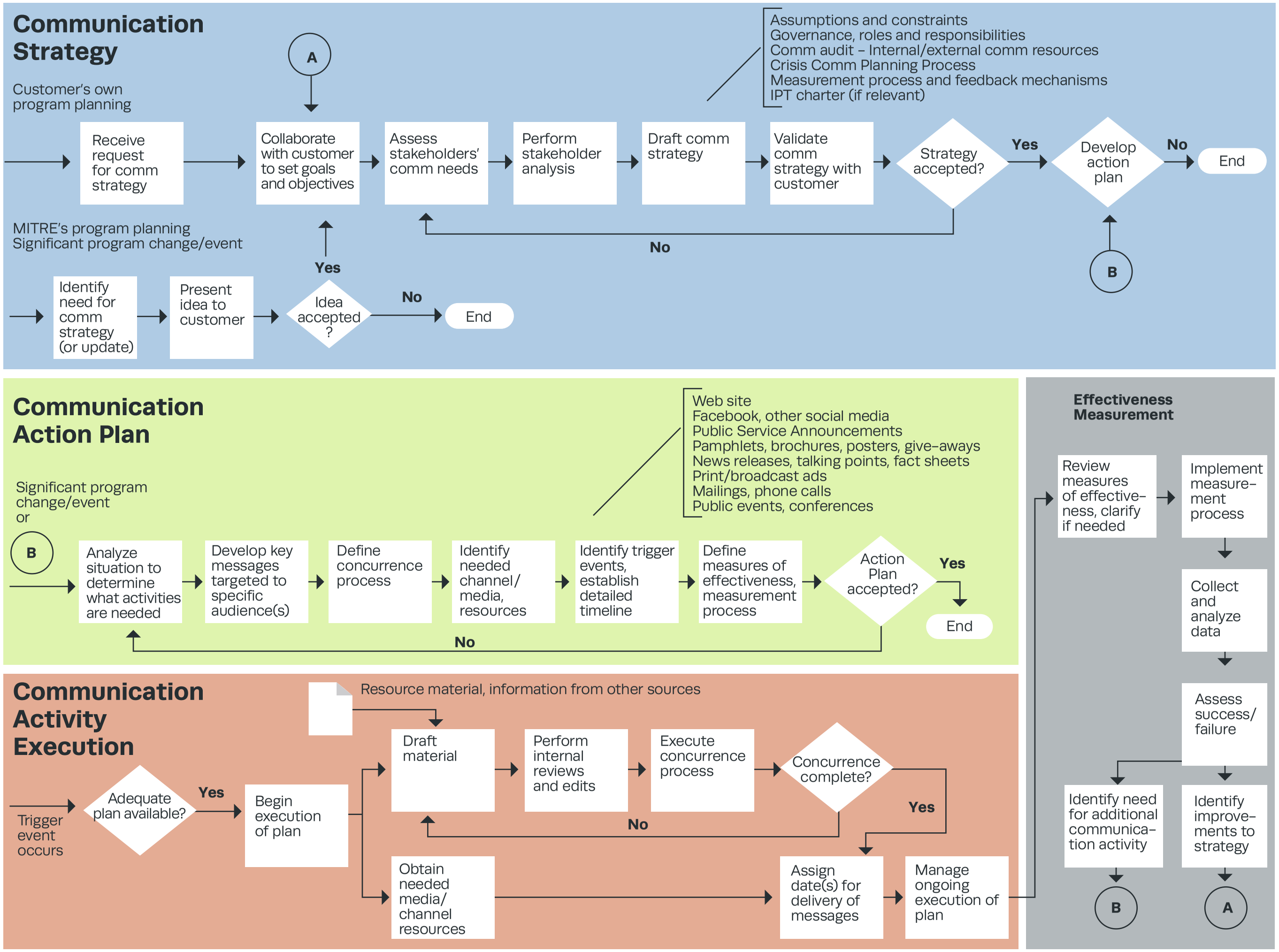

Communication Strategy
Customer's own program planning
Receive request for comm strategy
A
Collaborate with customer to set goals and objectives
Assess stakeholders' comm needs
Perform stakeholder analysis
Draft comm strategy
Validate comm strategy with customer
Assumptions and constraints
Governance, roles and responsibilities
Comm audit – Internal/external comm resources
Crisis Comm Planning Process
Measurement process and feedback mechanisms
IPT charter (if relevant)
Strategy accepted?
Yes
No
Develop action plan
No
End
B
MITRE's program planning
Significant program change/event
Identify need for comm strategy (or update)
Present idea to customer
Idea accepted ?
Yes
No
End
Communication Action Plan
Significant program change/event or
B
Analyze situation to determine what activities are needed
Develop key messages targeted to specific audience(s)
Define concurrence process
Identify needed channel/ media, resources
Identify trigger events, establish detailed timeline
Define measures of effectiveness, measurement process
Web site
Facebook, other social media
Public Service Announcements
Pamphlets, brochures, posters, give-aways
News releases, talking points, fact sheets
Print/broadcast ads
Mailings, phone calls
Public events, conferences
Action Plan accepted?
Yes
End
No
Communication Activity Execution
Trigger event occurs
Adequate plan available?
Yes
Begin execution of plan
Resource material, information from other sources
Draft material
Perform internal reviews and edits
Execute concurrence process
Concurrence complete?
Yes
No
Obtain needed media/ channel resources
Assign date(s) for delivery of messages
Manage ongoing execution of plan
Effectiveness Measurement
Review measures of effective-ness, clarify if needed
Implement measure-ment process
Collect and analyze data
Assess success/ failure
Identify need for additional communica-tion activity
B
Identify improve-ments to strategy
A

Figure 1. Communications Roadmap

2. Review measures of effectiveness and clarify, if needed.
3. Implement measurement process.
4. Collect and analyze feedback data.
5. Assess success of communication activity.
6. Identify need for additional communication activities or improvements.
7. Incorporate feedback from measurements to continuously improve the communication process and activities throughout the transformation.

Most management failures result from a failure to communicate somewhere along the line. Recognition of this need to communicate ought to be written into the job specifications of every chief executive and senior manager.
– Jacques Maisonrouge, former chairman, IBM World Trade Corporation

The exact form of communication needed during a transformation project is driven by a variety of factors: the sponsor's culture, the nature of the transformation, the communication channels available, and the time and resources available for communication activities. Consider the following key discussion points for developing sponsor buy-in for communication planning:

Explain the importance. A failed program is a waste of valuable funds, time, and reputation.

- Three out of four large IT programs fail to achieve their objectives.
  - Poor communication is a primary factor in one-third of failed programs [*PwC Mori Survey 1997*].
  - Good communication is a critical factor in 70 percent of successful programs.
- Communication planning reduces risk of project failure.
  - Accurate, truthful, and timely information replaces gossip and rumor and eases anxiety.
  - Key leaders will become champions if they understand fully the impact and benefits.
  - Employees who trust the communication process are more secure, focused, and productive, providing better service to constituents.

Explain the value. People are afraid of the unknown, but they'll support a project if they can see its value.

- Resistance to change is normal. In government agencies, resistance is caused in part by:
  - The graying workforce—nearly half of government employees are approaching retirement in the next five years. Some may lack the time to learn new processes or skills that a major long-term change may require.
  - Change fatigue—employees are exhausted by multiple (and often conflicting) initiatives launched by short-term appointees.
  - "Wait them out"—change is often imposed from above with little input from actual users, who may delay the change by simply waiting until the leadership changes.
- Focus on addressing "What's in it for me?"
  - Identify key stakeholders and their particular concerns and needs.

- Determine the specific benefits (and pain points) of the project for each stakeholder group.
- Communicate early, often, and clearly. Tell stakeholders what is going on, tell them why, tell them what they need to do, and specify the benefits for them.
- Set up feedback mechanisms and solicit stakeholder input to continuously review and improve the project.

References and Resources

1. Amin, A., November 2008, The Communication Key of Program Management, *PMWorldToday*, X (XI).

Additional References and Resources

Campbell, G. M., 2009, *Communications Skills for Project Managers*, AMACOM.

Cialdini, R., 2008, *Influence: Science and Practice,* 5th Ed., Boston, MA: Allyn & Bacon.

Dow, W., and B. Taylor, B., 2008, *Project Management Communications Bible*, Wiley Publishing.

Harvard Business School Press, 2003, *Business Communication*, Harvard Business School Publishing.

Hirsch, H. L., *Essential Communication Strategies for Scientists, Engineers, and Technology Professionals,* 2nd Ed., John Wiley & Sons.

Kendrick, T., 2006, *Results Without Authority: Controlling a Project When the Team Doesn't Report to You—A Project Manager's Guide.* American Management Association, AMACOM.

Kerzner, H., 2009, *Project Management Case Studies*, John Wiley & Sons, Inc.

Kliem, R. L., 2008, *Effective Communications for Project Management*, Auerbach Publications.

Perloff, R., 2008, *The Dynamics of Persuasion: Communication and Attitudes in the 21st Century*, 2nd Ed., Mahwah, NJ: Lawrence Erlbaum Associates.

Phillips, J. J., and W. F. Tush, W. F., 2008, *Communication and Implementation: Sustaining the Practice*, Pfeiffer.

Ruben, B. D., and L. P. Steward, May 2005, *Communication and Human Behavior,* 5th Ed., Boston, MA: Allyn & Bacon.

TRANSFORMATION PLANNING AND ORGANIZATIONAL CHANGE

Planning for Successful User Adoption

Definition: *Deriving from usability engineering and organizational change management, capability and technology transition strategies aim to increase the likelihood that users will adopt an application.*

Usability measures how intuitive, efficient, and task-enabling users think an application is.

Usability engineering refers to structured methods applied to achieve usability in user interface design during the entire application development life cycle.

Organizational change management is the art and science of moving an organization from its current state to a desired future state.

Keywords: *organizational change, transition strategies, usability engineering, user adoption*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to develop and recommend strategies to eliminate, reduce, and manage end-user reluctance to use newly deployed capabilities or technology or outright reject them. They are expected to define requirements for capability and technology transitions in requests for proposals (RFPs) and related acquisition documents and assess the quality of bidder responses as a key element of the source selection process. MITRE SEs are also expected to monitor and evaluate contractor capability and technology adoption efforts and the acquisition program's overall processes and recommend changes when warranted. Throughout the application development life cycle, MITRE SEs serving in project management, development, and oversight roles should emphasize the importance of transition plans in achieving successful end-user adoption.

Background

Everyone agrees that users and developers need to work together to build systems. The question is “how? [1]” In developing mission-critical applications that involve significant investment of time and resources, how can we ensure that the application will actually be used once deployed? Two primary factors impede user adoption of mission applications: (1) the lack of acquisition program and user mission management support, and (2) the failure to develop a tool as “seen through the user’s eyes [2].” While there is ample evidence that some applications developed and deployed by the government are never used or require massive redesign after roll-out, there is a dearth of baseline metrics for understanding the extent of the problem or developing “best practice” solutions. Thus, we still rely on intuition and anecdotal evidence to guide our understanding of the problems and solutions to improve outcomes.

Today’s standard practice for achieving user adoption of a new application is to follow the engineering development life cycle, define requirements, and establish user groups as sounding boards. But even with enthusiastic early adopters providing inputs, the number of times that users reject sponsor applications at the point of deployment does not inspire confidence in this approach. Many of these applications require years of subsequent releases to get them right, while others simply die a slow death as the funding runs out, having been deployed but never widely adopted.

This article describes strategies for stimulating user adoption of mission applications, including usability engineering to align the application with user tasks and organizational change management strategies to ensure the organizational readiness of the stakeholders to promote and accept change.

Strategies for Stimulating User Adoption

Usability Engineering

Usability measures how intuitive, efficient, and task-enabling an application is. Usability engineering refers to the structured methods applied to achieve usability in user interface design and process enablement. MITRE experience suggests that relying exclusively on user groups to derive requirements and engage users in an application development effort is not a promising path. User groups are necessary but insufficient for engaging users. They attract early adopters and technical enthusiasts, and while they provide valuable input, their limited representation quickly compromises the engagement process.

Usability engineering can fill the representation gap. It provides formal methods for identifying the different types of users, modeling the tasks they perform, and deriving usage conventions for the new application from those used in the existing software environment [3]. It employs structured methods such as field observations, task assessments, heuristic evaluations, and cognitive walkthroughs to obtain actionable feedback from representative users. Usability engineering provides a method for determining which users to engage and the kind of information needed to create usable designs, the optimum time to gather different types of user inputs in the application development life cycle, the most promising approaches to soliciting user inputs, and how to resolve disagreements among users. Further, it provides an objective means of measuring the progress toward establishing the usability of an application by developing and periodically assessing quantifiable usability requirements. Ideally, therefore, usability engineering begins early in the acquisition process and operates throughout the life cycle of application development.

Organizational Change Management

Organizational change management is a discipline that moves an organization from its current state to a desired future state. It derives its methodologies from the social and behavioral sciences, particularly organizational psychology, communication theory, and leadership development. The deployment of enterprise mission applications typically involves some type of change, such as the imposition of new workflows, business processes, quality standards, and/or metrics for measuring effectiveness. Addressing the organizational dimensions of deploying new technology is critical to establishing manager buy-in and engendering user adoption. For more information, see the other articles in the Transformation Planning and Organizational Change topic.

Some acquisition stakeholders interpret application transition narrowly as hardware and software transition. The focus of transition should be extended to include the broader organizational issues relevant to user adoption. This includes: (1) leadership engagement to involve business managers in promoting new technology and ensuring organizational readiness, (2) release strategies designed to optimize end user adoption, (3) communication strategies that prepare stakeholders for the coming change, and (4) training that equips end users to perform tasks with the new technology. The fifth element is usability engineering, as described above, which provides a quantitative means of ensuring the usability of an application as it evolves. In combination, these strategies address the factors that affect organization and site readiness [4] to embrace change in both processes and technology.

Best Practices and Lessons Learned

Build the right multi-disciplinary systems engineering team. Recognize that MITRE's contribution to acquisition, oversight, or development activities may call for usability engineers and organizational change specialists to be embedded in the acquisition and or systems engineering team. This is a form of the wisdom that says, "build teams with the skill sets needed to achieve a successful outcome."

Include a transition strategy in the RFP. In addition to detailing requirements, the RFP should include a transition strategy that delineates the role of government, contractors, and if applicable, federally funded research and development centers (FFRDCs), in stimulating user adoption of a new application. This will assure that the strategy has been discussed and agreed to by users and other stakeholders before issuing the RFP. Bidders should be asked to detail the methods and personnel they will employ in executing this strategy. The contractor strategy should be a major source selection criterion.

Convene a transition team on day one. Consider including the following team members: (1) contractor transition lead, (2) government transition lead, (3) mission-side transition lead, (4) usability engineering lead on development side, (5) independent assessor usability lead, (6) organizational change management lead, and (7) training lead. Consideration should be given to establishing a role for an independent usability assessor, in an organizational change capacity, or as the lead or co-lead of the transition team (in concert with or in lieu of a government team member). The team should monitor risk, measure progress, and steer the program toward developing an application that is quantifiably verified as usable. The transition team and program manager should develop and agree on a process for verifying transition-readiness.

Identify and prioritize user types. Define the different user types and rank their needs. If contention surfaces among requirements, schedule, and budget, this will ensure that necessary trade-offs are informed by mission priorities.

Continue usability assessment after an application is deployed. Continue to survey the rate of user adoption, assess ease of use, determine the effectiveness of training, etc. Usability engineering starts at program inception and continues after deployment to identify operational or field-specific problems [5].

Recognize "red flags" and address them early. Develop early indicators to alert you that the organization's change approach is likely to founder. These indicators may appear at the very beginning of an effort and continue through deployment. Recognizing the red flags and addressing them early can get you back on track. Here are some examples:

- **IT expects to lead transformational change for the mission:** IT can partner, IT can support, but IT cannot lead an organizational change initiative for the mission side. Without mission leadership, the effort to introduce transformational technology will fail.

- **Transition team is buried or marginalized:** We have all seen it. Program management pays lip service to user adoption when necessary to get through control gates or milestones. If management and the mission are not aligned and committed to supporting transition strategies, the ultimate outlook for application adoption is grim.
- **Expectation that transition will be handled by the developers:** Developers play a key role in building user-centric applications, but expertise in usability engineering and organizational change is fundamental to a successful process.
- **Increasingly greater reliance on training as the "cure-all":** A growing need for training is often an early indicator of a decline in usability and design quality. Avoid the temptation to rely on "training" as a workaround in lieu of building usable, intuitive systems.

Consider convening an enterprise-wide developer group. Involve developers throughout the application life cycle. In one program for a single sponsor, a monthly meeting was introduced where developers come together to harmonize interface designs and usage conventions across major enterprise applications under development. This is an initial effort in the early stages of implementation, but the expectation is that it will lead to more usable systems. While the early results appear promising, the ultimate impact of this strategy is still being assessed.

Expect resistance. Do not assume that project managers will be rewarded for advocating usability. Many organizations still reward project managers on the basis of their adherence to schedule and budget, regardless of adoption outcomes. In fact, the norm in many organizations is initial user rejection of applications, followed by years of subsequent releases to "get it usable." These dysfunctional practices are difficult to turn around, particularly when they are entrenched in organizational culture and reward systems.

Expect formidable challenges when asked to introduce transition planning at the end of the development life cycle. You may be called in after the fact when initial application transition has failed. With enough time, money, and expertise, it may be possible to determine what went wrong and how to correct it, but turning around a failed deployment is a daunting task.

Bottom Line

To stimulate user adoption of transformational technology, the systems engineering effort requires an interdisciplinary team that includes embedded experts in organizational change and usability engineering who can leverage relevant social and behavioral methods.

Adoption of new technologies requires a transition team that can develop an overarching plan, manage the activities as a coherent whole, and measure progress toward the development of a usable system. The team should begin these activities at program inception and continue throughout the acquisition phase and after deployment.

References and Resources

1. For a fundamental introduction to technology adoption, see Rogers, E.M., 1995, *Diffusion of Innovations,* 4th Ed., New York: The Free Press.
2. Adelman, L., P. Lehner, and A. Michelson, September 27, 2009, *Why Professionals Are Reluctant to Use Judgment and Decision Aids: Review of the Literature and Implications for the Development Process,* The MITRE Corporation.
3. MITRE has significant experience pairing developers with analysts in "small cells" to elicit requirements and build tools in real time for a discrete set of users. Organizations developing technology for a small group of users should consider adopting the "small cell" approach. See Games, R., and L. Costa, April 2002, *Better Intelligence Analysis Through Information Technology—Lessons Learned from the Analysis Cell Initiative.*
4. For more on usability engineering, see Bradley R., T. Sienknecht, M. Kerchner, and A. Michelson, October 2005, Incorporating Usability Engineering into Application Development, draft briefing; May 2007, Incorporating Usability Engineering into Application Development, draft briefing; and Bradley, R., E. Darling, J. Doughty, and T. Sienknecht, September 4, 2007, Findings from the Agile Development and Usability Engineering TEM, MITRE briefing.
5. For an excellent example of post-deployment usability engineering, see Kerchner, M., June 2006, "A Dynamic Methodology for Improving the Search Experience," *Information Technology and Libraries*, Vol. 25, No. 2, p. 78–87.

Enterprise Governance

Definition: *MITRE systems engineers are expected to develop a broad understanding of the policy making, capability management, planning, and other business functions of their sponsor or customer enterprise that either influence or are influenced by systems engineering. They are expected to recommend and apply systems engineering approaches that support these business enterprise functions.*

Keywords: *COI, customer focus, governance, outcomes, policy, standards, strategy*

MITRE SE Roles and Expectations

Taken together, the different levels of customer governance define what the government is attempting to achieve, how they intend to go about it, and why they are attempting to do so. MITRE systems engineers (SEs) are expected to understand the governance of the programs they support as well as the larger enterprise in which the programs are embedded. This understanding is important even if MITRE staff are not directly contributing to or influencing the day-to-day workings of that governance. When MITRE SEs understand how their program fits into its governance context, they can factor that knowledge into making technical recommendations. This directly affects MITRE's ability to provide systems engineering value and impact.

The expectations for MITRE SEs depend on the position and role in the sponsor's governance organization:

For MITRE staff executing tasks to support specific programs, our influence is usually focused on our customer's program outcomes. This requires an understanding of our immediate customer's objectives and desired outcomes so we can help the customer achieve them. In doing so, we should always provide independent, unbiased recommendations. In formulating recommendations, it is critically important that MITRE SEs take into account the other levels of the customer's governance organization, whether or not the immediate customer does so. On occasion, this may mean that we do not always agree with the immediate customer's direction. But it does require that we explain how consideration of the other levels of governance factored into our recommendations.

- MITRE staff also play a role in helping the customer define or refine business processes, such as technical or systems engineering aspects of portfolio management. For mid- and senior-level MITRE staff, our role often involves recommending how to apply engineering analysis, advice, processes, and resources to achieve desired portfolio outcomes. Understanding the interconnections and dynamics across the different levels of the customer's governance structure is important to providing thoughtful, balanced recommendations.
- Most MITRE staff probably have little influence in directly shaping enterprise behavior and standards on a continuing basis. However, for staff that participate in enterprise-level activities, such as representing their programs in communities of interest (COIs), participating on standards boards, helping define policy at high levels within the government, etc., the influence can be broad and far reaching. When in such a role, MITRE staff are expected to coordinate extensively across the corporation, other FFRDCs, academia, and industry to ensure that all affected programs and other stakeholders are aware of the activity and can provide input to shape products or decisions. MITRE contributions should consider all aspects of the problem, provide an enterprise perspective, be product and vendor-neutral, and anticipate future missions and technologies.

Context

MITRE's systems engineering support exists in a complex political, organizational, and process-driven environment. Many of the actions and behaviors of our customers are motivated by this environment. Although MITRE's work is focused on technical aspects of the systems and enterprises, it is essential that our systems engineers be aware of, understand the effects of, and navigate effectively within the governance structure of how systems and enterprises are acquired and managed. Whereas the other topics in the Enterprise Engineering section of the SEG focus on the more technical aspects of our systems engineering support in the project's enterprise, the Enterprise Governance topic addresses the business and policy environment of our projects.

Governance of programs, projects, and processes can be envisioned as operating at three different, interconnected levels of our clients' organization:

- **Program level:** In general terms, success at this level is defined by meeting the goal of delivering a system that meets specified, contracted-for performance, price, and schedule parameters. Program-level decisions, directions, and actions align with that view of success and influence the expectations of systems engineering provided at that level.
- **Portfolio level:** At his level, the focus shifts to making trades among a collection of programs to achieve capability-level outcomes. The trade-offs balance various criteria, including the importance of capabilities to be delivered, likelihood of program success, and expected delivery schedule within constraints, like availability of funding and capability operational need dates. Portfolio-level decisions can result in programs being added and accelerated, cut back and slowed, deferred, or even cancelled.
- **Enterprise level:** Interventions at his level shape and change the environment (or rules of the game) in which programs and portfolios play out their roles and responsibilities to achieve enterprise-wide outcomes, like joint interoperability or net-centricity. Often this is achieved through department or agency-wide policies or regulations that rarely directly control, manage, or decide the fate of specific programs or portfolios. Instead, they indirectly influence programs and portfolios to stimulate variety and exploration of technologies, standards, and solutions or reward, incentivize, or demand uniformity to converge on common enterprise-wide approaches.

These levels of governance and their interactions are not unique to government departments and agencies. An example of a similar construct is the U.S. economy. Commercial companies produce consumer goods, like automobiles, to gain market share by offering products with competitive price-performance potential. Large commercial enterprises (e.g., General Motors [GM]) maintain portfolios of these products and expand or contract them based on market analyses and economic forecasts. Examples include GM closing production of its Saturn line of automobiles and selling off Saab. Of course, GM's goals are quite different from a government organization managing a capability portfolio to support government operations, but the essential governance considerations, analyses, and decision making are remarkably similar.

Best Practices and Lessons Learned

View governance as scaffolding, not prison bars. Think of governance as a mechanism for navigating the solution space for the problems we are attempting to solve for our customers, rather than as a set of restrictive constraints that inhibit our freedom. Use the governance principles as a context for guiding your recommendations. Leverage and use the governance concepts to support your recommendations when they align (as they usually will).

Finding the right balance point. As you conduct your tasks, ask yourself whether the direction of your results (products, recommendations, etc.) is consistent with the enterprise's governance structure, including the customer's business processes, broader COI, and top-level policies and standards. If not, consider whether the results or the current position defined by the governance is more important to the program and enterprise you are supporting.

Be willing to challenge governance if you decide it is not valid or is detrimental to achieving desired results. Although high-level governance practices, such as policies or standards, are created with the intent of being applicable across the enterprise, generally those practices cannot account for every possible situation. Strike a balance between the governance practices that work for most parties and situations with those appropriate for your program and enterprise. Use your engineering judgment, combined with your broad understanding of the governance practices (and how they are applied and why they are beneficial), to determine that balance.

If governance practices need to be changed, consider how to augment, adjust, or refine existing guidance while satisfying local objectives, rather than recommending dramatic changes that may be almost impossible to achieve. When recommending changes, ensure that the intent of the governance concepts is honored and that the proposed revisions are applicable to and suitable for programs and enterprises beyond your local environment or situation. Enlist support from peers and management—first within MITRE, and then with the sponsor—to effect the changes desired. Develop a strategy early on about how to accomplish the governance changes, accounting for stakeholders, their motivations, and how to achieve win-win results.

Articles Under This Topic

"Communities of Interest and/or Community of Practice" provides advice on working in groups that collectively define items like interoperability concepts.

"Standards Boards and Bodies" provides best practices and lessons learned for MITRE staff participating on technical standards committees shaping technical compliance in programs and systems.

"Policy Analysis" discusses MITRE support to decision making in a multi-stakeholder, multi-objective environment.

Definition: *A Community of Interest (CoI) and/or Community of Practice (CoP) is a group of people operating within or in association with a client, customer, sponsor, or user in MITRE's business realm or operating sphere of influence for the purpose of furthering a common cause by sharing wisdom, knowledge, information, or data, and interactively pursuing informed courses of action.*

Keywords: *community of interest, community of practice, group dynamics, information exchange, mission success, mutual trust, shared goals, systems integration, terminology, user involvement*

ENTERPRISE GOVERNANCE

Communities of Interest (COI) and/or Community of Practice (COP)

MITRE SE Roles and Expectations: MITRE systems engineers are expected to participate in CoIs or CoPs associated with their projects. This will assist in harmonizing domain terminology, exchanging pertinent information, and looking for and acting on issues and opportunities in their project's enterprise. MITRE staff in CoP or CoI settings are expected to bring the corporation to bear by providing the greatest value to our clients, customers, sponsors, or users, in conjunction with other government contractors.

Characteristics of CoIs and CoPs

The terms CoI and CoP are sometimes invoked interchangeably, but there are distinctions:

- **Communities of Practice** are "groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in the topic by interacting on an ongoing basis." They operate as "learning systems" or "action systems" where practitioners connect to solve problems, share ideas, set standards, build tools, and develop relationships with peers and stakeholders. A CoP is typically broader in scope and tends to focus on a common purpose, follow-on actions, and information exchanges. CoPs can be both internal and external to MITRE, including various government, industry, academia, and MITRE participants.
- **Communities of Interest** are typically narrower in scope and tend to have a specific focus, such as information exchange. COIs typically tend to be a government organization approach (particularly in the Department of Defense [DoD]) to bring together individuals with common interests/references who need to share information internal to their community. They also need to provide an external interface to share with other communities (e.g., allowing a community-based loose coupling and federation as highlighted in other Enterprise Engineering section articles).

The MITRE business realm, program, or project context may determine which term is used more prevalently according to these characteristics.

The characteristics of CoIs and CoPs are discussed below in terms of social interactions, operations, longevity, and commitment.

Operations

A CoP may operate with any of the following attributes:

- Some sponsorship
- A vision and/or mission statement
- Goals and/or objectives
- A core team and/or general membership
- Expected outcomes and/or impacts
- Measures of success
- Description of operating processes
- Assumptions and/or dependencies
- Review and/or reflection

Often CoIs span similar organizations (e.g., DoD, particularly when there is a common interest in an outcome).

Individual members may be expected to:

- Support the CoP through participation and review/validation of products

- Attempt to wear the "one hat" associated with the CoP while maintaining the integrity and autonomy of their individual organizations.
- Participate voluntarily with the blessing of their organizations that determine their level of participation and investment.

Sponsoring organizations might provide a nominal budget needed to participate in CoIs/CoPs, make presentations at external organizations, or support meetings of the core team. Thus MITRE staff participating in CoPs must be mindful of the time and effort they contribute, and ensure that their participation is an appropriate and justifiable investment of project resources.

Longevity

The "practice" part of CoP relates to the work the community does. This includes solving common problems, sharing ideas, setting standards, building tools, and developing relationships with peers and stakeholders. Collective learning takes place in the context of the common work. These groups learn to work not so much by individual study, lectures, etc., but by the osmosis derived from everyone working together—from experts to newcomers—and by "talking" about the work. This provides value to all organizations represented. MITRE participants should attempt to contribute their best ideas to the CoP and bring back good practices to share with their project teams and colleagues.

Like many other communities, a CoP grows based on the increasing benefits individuals or organizations accrue from participating in the activity. Sometimes these rewards include personal satisfaction in contributing and being recognized for adding value. A CoP that has a positive impact by helping to solve important problems not only retains a substantial percentage of its members, but attracts new ones.

Social Interactions

A CoI or CoP can operate in various interpersonal modes, including face-to-face or via video/audio teleconference, telephone, email, and website access devices. MITRE has been operating in all these modes for some time; it is becoming increasingly immersed in the virtual environments. It behooves us to become familiar and adept with the newer, more pervasive and effective methods.

Important Topics Relevant to CoIs and CoPs

Terminology

One important goal of any community is to achieve a shared understanding of terminology, particularly community-specific terms of art. It is not unusual for different stakeholders in a

CoI/CoP to start with different meanings for a given word, or different words for something with a common meaning. Because MITRE is frequently in the position of providing technical support to different constituents of a CoI/CoP, we are in a position to assess whether a common understanding of a term is important to achieve and, if so, how to achieve that harmonization. For example, it is critical that the term "identification" has a commonly understood meaning between the military tactical situation awareness community and the intelligence analysis community when the two communities are operating together.

Information Sharing

One of the primary CoI/CoP functions is to share information that is important to the common purpose of the CoI/CoP. There are forces at work in organizations that may discourage information sharing, either explicitly or implicitly. The reasons for this are many. Some are legitimate (e.g., sharing has the potential to compromise classified information or threaten network security) while others are an artifact of organizational cultures that see the retention of information as a form of power or self-protection.

Trust

Good relationships are built on interpersonal trust. In the context of CoIs/CoPs, trust assumes two forms. First, information provided by an individual must be true and valid (i.e., the individual is viewed as a competent source of information). Second, a trustworthy person is dedicated to the goals of the CoI/CoP and treats others with respect. Trust is an important ingredient in the facilitation of information sharing.

Group Dynamics

Effective participation and operation within a CoI/CoP is highly correlated with good interpersonal skills in group settings. This requires an awareness and understanding of human motivation and behavior.

CoI Lessons Learned

Value and Focus

- **Purpose:** Start with a clear purpose and informed understanding of the requirements. Lack of clearly defined requirements can cause restarts. Define the scope early, work closely with the formation teams to ensure all necessary information is included, and prevent "requirements creep."
- **Passion:**. Known consumers with known needs are important for CoI success. Programs of record that have an imperative to

deliver capability to the warfighter and are dependent on CoI products to do so can be used to drive toward results.

Strategy

- **Objectives:** Define the terminology, goals, and constraints, and understand the problem and objectives so people are willing to participate. Ensure there is a well-defined purpose addressing a scoped problem relevant to the participants and tackled in achievable increments/spirals to adapt to changing needs. Select a scope permitting delivery of useful results within 9 to 12 months. Try to adopt a common model and avoid generating unique vocabularies and schema across domains to prevent an "N-squared problem" of communication among participants. System builders are usually important contributors to CoI vocabulary work and should be encouraged to drive the common vocabulary activities. Most CoI efforts do not have time to create or learn large new vocabularies, so leverage past efforts.
- **Stakeholders:**. Address cross-organizational and cultural challenges through structure, partnership, and transparency. Any issues and competing agendas need to be addressed directly, seeking common ground and win-win solutions. Institutionalize the CoI through creative friction and an adaptable system of rewards/incentives.
- **Invite key stakeholders to help ensure broader acceptance of results:** Identify organizations willing to contribute early in the process and those with a vested interest in the outcomes. It is better to get diverse inputs to surface showstopper issues early, so they can be dealt with appropriately as the work progresses.

Governance

- **Leadership:** Ensure that there is strong leadership and commitment to success. Both attributes are important to keep the team engaged and moving in a common direction. There is no substitute for governance, self-policing, and personal relationships.
- **Commitment:** Prepare for long-term commitment. CoIs are nontrivial and require significant levels of participation over time. This has the potential for significant unfunded costs to support and implement. Assess and continually re-evaluate the return on investment of MITRE's participation in the CoI.
- **Procedures:** Each CoI must establish its own operating procedures. Though other CoI procedures could be used as a basis, each CoI needs to tailor its norms and procedures to the organization (and culture) and objectives.
- **Have a set of exit criteria:** Develop a set of criteria and exit strategy for disbanding the CoI (or for MITRE to cease participation), using the CoI objectives and our intended role.
- **Limit attendance to one or two representatives per organization/program:** Try to limit attendance to key players (e.g.,

an authoritative manager and a technical expert).

- **Limit teleconferences to preparing for meetings or reviewing status:** Face-to-face meetings are required to get the work done. Teleconferences have limited benefit for working through complex issues.
- **Have important tasks and announcements distributed by a high-ranking leader to those with authority:** This tends to get people's attention and increases the level of cooperation. For example, official "taskers" need to be sent by a government representative to other government representatives when many of the CoI players are contractors.
- **Have fewer but longer meetings:** This improves the chance of retaining the same players and helps eliminate the problem of restarting and retracing steps and agreements made at previous meetings for the benefit of new players.
- **Take real-time minutes to ensure agreement on issues, results, and action items:** Take minutes and document significant happenings as they occur. This provides a tangible track record that helps prevent disagreements later.

Conclusion

As you participate in the CoI/CoP process, leverage the lessons learned in this article and identify additional ways to enhance the CoI/CoP efficiency and effectiveness. Equally important, share the lessons learned and best systems engineering practices that you experienced through participating in CoIs/CoPs.

References and Resources

Advanced Distributed Learning, "Simulation and Training Community of Practice," accessed January 22, 2010.

Generation YES Blog—Thoughts About Empowering Students with Technology, accessed January 22, 2010.

"Knowledge Management," accessed January 22, 2010.

Land Grant University CoP, "Applying to Become a Professional Development Community of Practice," accessed January 22, 2010.

"Leveraging the Corporation," MITRE Project Leadership Handbook, The MITRE Corporation.

Medric, L., February 9, 2007, "Communities of Practice (CoP)—An Overview and Primer," CEM Forum.

ENTERPRISE GOVERNANCE

Standards Boards and Bodies

Definition: *In many instances, MITRE's systems engineering and subject matter expertise is brought to bear in helping committees produce industry standards. Industry standards typically require the consensus of the committee's members, which may include representatives from government or industry or both. MITRE's contributions include direct technical contributions, managing committees and their documents, and helping to moderate negotiations between committee members to bring about consensus.*

Keywords: *AIEE, ASE, consensus documents, IEEE, negotiations, RTCA, standards*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) need to understand the objectives of the standards body that is producing the standard, typically articulated in Terms of Reference for the committee. They should ensure that the goals of the standards committee, the MITRE work program, and sponsor are in alignment. SEs are expected to bring expert technical analyses and discipline to the standards process by providing objective data relevant to the topic of standardization.

MITRE Interest

MITRE's interest in standards board participation is in establishing the best standards to further technology and industry implementation to improve interoperability across government capabilities. Participation in standards bodies provides opportunities to bring the world to bear on our customers' problems, and collaborate with other FFRDCs, industry, and academia.

Background

Standards committees have historically provided standards that allow for compatibility of equipment produced by different vendors, or that provide for minimum safety of equipment or devices. For example, the Radio Technical Commission for Aeronautics (RTCA) develops standards for aircraft equipment that allow multiple vendors' equipment to be cross compatible within the aircraft, from aircraft to ground systems, and between neighboring aircraft. Another example is minimum requirements for electrical equipment.

Standards bodies are typically attended voluntarily by participating government and industry organizations that send experts on standardization. Government employees may work directly with industry in developing the standards. Usually standards bodies' meetings are open to participation and to the public.

Most standards are agreed upon by consensus; that is, all participating organizations agree to the requirements represented in the standard. Arriving at a consensus can be challenging and time-consuming, which is a principal reason why standards sometimes take substantial time to produce. MITRE's expertise and objectivity can be key to brokering consensus.

Government Interest and Use

Many U.S. federal, state, and local government agencies depend on voluntary consensus standards. In many cases, the U.S. government relies on standards bodies to provide input to potential government rules and regulations. RTCA, for example, functions as an advisory committee to the Federal Aviation Administration (FAA) under the U.S. Federal Advisory Committee Act of 1972 [1]. RTCA standards, when accepted by the FAA, may become the basis for FAA Technical Standard Orders, which govern the requirements for equipment manufacture, or FAA advisory circulars, which provide advice on equipment installation, usage, etc.

An agency may adopt a voluntary standard without change by incorporating the standard in an agency's regulations or rules. Depending on the relationship of the standard body to the government, the government may adopt the standard with certain exceptions. In other generally exceptional cases, the government may ignore the standard outright. Under the U.S. Federal Advisory Committee Act, for example, a voluntary consensus standard is submitted to the government as advice, and the government is under no obligation to accept that advice. MITRE systems engineers can play an important role in brokering agreements between government and industry to ensure that the standards are accepted and utilized by government agencies [2].

In some cases, standards become the basis for rulemaking by the government. In this situation, the government will first propose the rule in the Federal Register as a notice to the public, known as a Notice of Public Rulemaking. At this stage, the public is invited to comment on the proposed rule. All comments must be considered by the government in advance of issuing a final rule. The government's response to the comments becomes a matter of public record. The relevant standards may form a substantial basis for the final rule; in some cases, a standard may be one accepted means of compliance, may be the basis for guidelines for compliance, or voluntary compliance with a given standard may be accepted in lieu of formal rulemaking [3].

The U.S. federal government updates the Code of Federal Regulations (CFR) once per year. The CFR contains all the rules published by the U.S. federal government. The CFR is divided into various areas of regulation [4].

Best Practices and Lessons Learned

Objectivity is paramount. MITRE must act, and be viewed, as an objective participant, since our goal is to be able to moderate negotiations. Committee participants should make sure that all perspectives are considered fairly, even though some perspectives may conflict with our sponsor's point of view. MITRE's role is to bring objective analysis to the table for all parties to consider. It is highly desirable to bring analytical results to the conversation to inform the discussions. In lieu of analytical data, objective expert opinion should be clearly articulated.

Bring the best expertise to the table. Committees are usually public forums in which MITRE's reputation and credibility are at stake. Most organizations tend to staff committees with their most senior and knowledgeable staff; MITRE should do no less. Specific subject matter expertise should be brought into conversations whenever appropriate; key staff should be on call to serve in these roles.

Involvement in committee leadership is an asset. One way to demonstrate MITRE's influence and objectivity more effectively is for MITRE participants to be involved in the committee leadership. MITRE roles have varied from high-level leadership positions in the overall organization (e.g., leading the RTCA Program Management Committee), leading special committees, leading working groups under the larger committees, and taking on the role of committee or working group secretary. All of these types of leadership roles reflect well on the company and put MITRE in a position of regard and influence.

Hold the pen. It might seem like a tedious job, but volunteering to manage the standards document puts MITRE in an effective position to help assume a key responsibility for the development of the standard. The book manager may be personally responsible for significant textual input to the standard. In addition, the book manager is responsible for coordinating inputs from various authors, managing configuration control, incorporating updates to the document, and ensuring that all comment resolutions are implemented as agreed.

Standards development should be a disciplined process. There should be clear, agreed procedures for running meetings with a leader who can moderate the conversation such that all voices are heard while progress and decisions are made. Documents should be developed with a clear configuration management plan. After a rough draft, documents should be reviewed and a record of comments and their dispositions, usually in the form of a comment matrix, should be maintained. A written record of proceedings is essential so that issues that have been discussed and dispositioned are not reopened.

Work difficult issues or specific subtasks in smaller subgroups. A key manner in which to accelerate the standards development process is to assign small groups to work out and agree on key technical points. In some instances, a small, ad hoc group may be formed to make recommendations on a specific issue. In other cases, more formal, long-term subgroups may be formed to draft whole sections of a standard. In any case, a "divide and conquer" approach is usually more effective in bringing acceptable proposals back to a larger group, rather than having a large group debate a given topic.

References and Resources

1. The Federal Advisory Committee Act, October 6, 1972, and the Federal Advisory Committee Act Amendments of 1997, December 17, 1997.
2. ANSI, "U.S. Government Use of Voluntary Consensus Standards," http://www.standardsportal.org/, accessed February 22, 2010.
3. Office of Management and Budget, February 10, 1998, "OMB Circular No. A-119: Federal Participation in the Development and Use of Voluntary Consensus Standards and in Conformity Assessment Activities."
4. GPO Access, "U.S. Code of Federal Regulations," accessed February 22, 2010.

Definition: *Policy analysis is a disciplined process to help people make decisions in situations of multiple objectives and multiple perspectives.*

Keywords: *decision making, policy, policy analysis*

ENTERPRISE GOVERNANCE

Policy Analysis

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the role and implication of policy in our customers' activities and how systems engineering relates to it. MITRE SEs are expected to know the basic characteristics of good policy analysis, so they can constructively collaborate with policy analysts on questions and issues arising at the boundary of systems engineering and government policy.

Government Interest and Use

Within the United States government, very few important decisions are made by a single individual. Congress and the Supreme Court make decisions by voting. While most executive branch decisions are made by an individual senior official, the decision is normally the result of a deliberative process in which numerous people with diverse expertise or diverse interests offer advice that it is imprudent to ignore. The output of a good policy analysis is a set of questions regarding priorities or a set of options among which to choose, along with the major arguments for each competing priority or the major pros and cons of each option. Either option will help organize the interactions that lead to choosing a course of action.

Systems engineering can be viewed as a process for arriving at a solution that represents an acceptable balance among multiple objectives. Traditionally, systems engineering typically presumed that the objectives and relevant operational constraints can be defined, and that the extent to which any given outcome meets a given objective can be quantified. When these conditions exist, systems engineering can usually arrive at a "best," "correct," or "optimal" design solution. Systems engineering also can derive the requirements for various subsystems on the basis of the overall system design, including the requirements that each subsystem must meet to interact properly with other subsystems. In contrast, policy analysis, when done well, leads to courses of action that may not be the "best" from any one perspective, but are "good enough" for enough players to win the necessary political support to move ahead. New forms of systems engineering are adopting this "good enough" solution perspective, particularly in large-scale enterprise settings. Indeed, a good policy analysis may be used by individuals who disagree with the government's objectives; in this case, one individual may conclude that the policy analysis shows option B is best while the other concludes that the same policy analysis shows option D is best, and they both agree that either option is acceptable.

Systems engineering and policy analysis must account for costs and affordability. An elegant engineering solution that the customer cannot afford is useless; so too is a policy option that would make many people happy, but at a prohibitive cost. Therefore, careful efforts to estimate the cost of a particular option and the risk that the actual cost may exceed the estimate are necessary for systems engineering and policy analysis. Engineers who design products for commercial sale are familiar with the concept of "price points," and a manufacturer may wish to produce several products with similar purposes, each of which is optimal for its own selling price. In the case of systems engineering for the government, it may be necessary to conduct a policy analysis to determine how much the government is willing to spend, before conducting a systems engineering analysis to arrive at the technically "best" solution at that cost level.

Best Practices and Lessons Learned

Especially rigorous quality assurance. Policy analysis at MITRE poses a special concern. The missions of MITRE's federally funded research and development centers (FFRDCs) are systems engineering and research, while policy analysis is the mission of other FFRDCs. At times, it is completely appropriate for MITRE to conduct policy analysis. MITRE has excellent policy analysts on its staff, but it falls outside the mainstream of our work. Thus, it important that all MITRE policy analysis delivered to the government is of high quality. If a MITRE policy analysis is substandard, we have few resources to fix the problem and are vulnerable to the accusation of taking on work that is outside our sphere of competency. Therefore, any MITRE policy analysis intended for delivery to the government typically requires a degree of quality assurance beyond our routine practices.

The technical-policy boundary—know and respect it. Some MITRE work requires policy analysis as a deliverable to our government sponsors (i.e., the sponsors ask us to provide analytical support for government policy-making). At times, MITRE conducts policy analysis for internal consumption only. This helps MITRE understand the multiple perspectives and objectives of our sponsors so that our technical work can be responsive to "real" needs that they may be precluded from expressing in official documents. Finally, MITRE is sometimes asked to support a government policy process by providing technical analysis that narrows the scope of the government's disagreements; the task of "taking the technical issues off the policy table" requires that MITRE staff sufficiently understand policy analysis to assure our technical analysis stops where true policy analysis begins.

Policy analysis basics for systems engineers. There are a few basics that characterize good policy analysis. MITRE SEs should be familiar with them, so they can constructively collaborate with policy analysts on questions and issues arising at the boundary of systems engineering and government policy. These are summarized in the order in which they appear during the course of a policy analysis:

- **Transform a situation into one or more issues:** The analysis must identify the policy decisions that are most appropriate for the situation. Figuring out what questions to ask is the most critical, and often the most difficult, part of the analysis. Asking the right questions is what transforms a "messy situation" into an "issue" or a "set of issues." When policy analysis is being performed for an identifiable customer, it is of little use unless the analysis is framed in terms of decisions that the customer has the authority to make—or perhaps decisions that the customer's boss or boss's boss has the authority to make, provided that the customer has a charter to go to the boss and say, "I can't do my job until you make this decision."
- **Create executable options:** The analysis must identify options for each decision. This is where policy analysis can be genuinely creative, even while remaining rigorous. There are many options in a typi-

cal government policy dilemma; however, the number of options that the policymakers can seriously consider is small. A senior government official looks for an option that will meet the most important objectives, can be implemented with the resources available, and will attract support from enough other perspectives to command a majority vote or support from a preponderance of advisers. A good set of options: (a) are responsive to the issues posed (see the previous bullet): (b) could be implemented, if chosen; and (c) none of the important players in the decision process will react by saying "none of the above."

- **Options have advantages, disadvantages, and uncertainties:** The analysis must identify the advantages, disadvantages, and uncertainties associated with each option. This is a straightforward process. However, if the analysis is to be credible, one must carefully state the pros and cons in ways that are recognized as accurate by those whose views they portray. For example, an analysis of an option for sharing extremely sensitive intelligence with an ally should state the pros in language that might be used by a proponent of this option, and the cons in language that might be used by an opponent. Otherwise, the product will be viewed as advocacy, not an analysis.
- **Strategies for reducing uncertainty:** Sometimes an analysis, having identified uncertainties that make it difficult to choose an option, may propose a strategy for reducing the uncertainties. Of course time reduces some uncertainties, and a serious effort to gather additional information will require time. Delaying a decision often permits a bad situation to become worse. Much of the art of the statesman is sensing the moment to make a difficult decision. When a policy analyst chooses to propose a strategy for reducing uncertainty, the analyst is helping the decision-maker understand how much time would be required to obtain additional information or understanding, and thus make a good judgment about when to decide.
- **Identifying additional options, if needed:** Sometimes if an analysis failed to identify an acceptable set of options, it may propose a strategy for identifying additional options. Such a strategy could be a targeted research program or consultation with other organizations that have not participated in the process.
- **Decision-making strategies:** Finally, the analysis may identify a strategy for arriving at a decision. In some circumstances, this is not necessary if the strategy is obvious; in other cases, some or all of the options may require concurrence of others or a process that is unusual in some way.

As is the case with many MITRE services and products, a policy analysis may contain extensive data and argumentation that the actual decision-maker will never read or hear. The executive summary of a paper and the first few slides of a briefing must clearly convey the information that the decision-maker should learn and understand, while the body of the paper and the extensive back-up slides in the briefing provide credibility to the message and a means for staff to check the validity of summary statements they find surprising. Therefore, it is highly desirable that the executive summary or the summary slides be well written. In contrast, the segments providing detail must be checked carefully for clarity and accuracy, but need not be models of graceful prose.

References and Resources

Allison, G. T., and P. Zelikow, 1999, *Essence of Decision: Explaining the Cuban Missile Crisis*, 2nd Edition, New York: Longman.

Conklin, J., Winter 2009, "Building Shared Understanding of Wicked Problems," *Rotman Magazine*.

Rittel, H., and M. Webber, 1973, "Dilemmas in a General Theory of Planning," *Policy Sciences*, Vol. 4, Elsevier Scientific Publishing Company, Inc., Amsterdam, pp. 155–169.

"Trade-off and Engineering Analysis," MITRE Project Leadership Handbook, The MITRE Corporation.

Wildavsky, A., 1979, *Speaking Truth to Power: The Art and Craft of Policy Analysis,* Little, Brown.

Wildavsky, A., 1988, *The New Politics of the Budgetary Process*, Scott, Foresman & Co.

MITRE FFRDC Independent Assessments

Definition: *An independent assessment is a tool that can be used at any point in a program life cycle to provide insight into progress and risks.*

Keywords: *accident investigation, audit, baseline assessment, independent expert review, independent technical assessment, red team, SCAMPI appraisal, Tiger Team*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to lead or participate in independent assessment teams, particularly when program processes are being evaluated or there are concerns about program progress or contractor performance [1]. MITRE SEs are expected to apply strong domain and technical expertise and experience and perform with objectivity consistent with the FFRDC role.

Context

An Independent Assessment is a team activity in which the team leader and team members are *not* members of the organization being assessed. This allows the team to more readily fulfill its charter objectively and without conflicts of interest. Ideally, assessments are proactive and intended to provide an early look into what potential problems may be on the horizon in time to take action and avoid adverse impact to the program. For example, an assessment may be used to take an early look at the challenges associated with technologies and provide feedback to the technology development strategy. In other cases, an assessment might be intended to assess progress with a process improvement framework, such as CMMI (Capability Maturity Model Integration) [2], or may be intended to examine causes of concerns with program performance.

In MITRE's work, independent assessments are known by several names, including:

- Independent Reviews
- Red Teams
- Blue or Green Teams
- Technical Assessments—Tiger Teams
- Appraisals (against a model or standard)
- Independent Expert Reviews (IER)
- Audits and Compliance Assessments
- Accident Investigations

For related information, see the article, "Planning and Managing Independent Assessments" in this topic, and "Data-Driven Contractor Evaluations and Milestone Reviews" and "Earned Value Management" in the topic Contractor Evaluation in the Acquisition Systems Engineering section.

MITRE SEs are frequently asked to lead and participate in independent assessments because the characteristics of an FFRDC, as chartered under the Federal Acquisition Regulation (FAR) 35.017 [3], promote independence, objectivity, freedom from conflicts of interest, and technical expertise. These characteristics support the management goals of the assessment team.

For example, the FAR describes the special relationship between a sponsor and its FFRDC [3]. The FFRDC:

"... meets some special long-term research or development need which cannot be met as effectively by in-house or contractor resources ... to accomplish tasks that are integral to the mission and operation of the sponsoring agency."

"...has *access, beyond that which is common to the normal contractual relationship*, to government and supplier data, including sensitive and proprietary data, and to employees and facilities."

"...conduct[s] its business ... to *operate in the public interest* with objectivity and independence, to be free from organizational conflicts of interest, and to have full disclosure of its affairs to the sponsoring agency."

Best Practices and Lessons Learned

MITRE's assessment teams have experienced some challenges, including [4]:

- Sponsor oversight has been delegated, leading to a lack of clarity about what needs to be accomplished.
- The review is additional work and seen as a lower priority.
- Members of the organization being reviewed are not able to participate as planned.
- Subjects of the review are not prepared for the review.
- Objective evidence is difficult to locate.
- Appraisal team working space is rarely available.

MITRE practitioners have found that these and other challenges can be mitigated through communication and following the process described in the "Planning and Managing Independent Assessments" article in the SEG.

References and Resources

1. Independent Technical Assessments.
2. The MITRE Institute, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1, p. 39.
3. Standard CMMI® Appraisal Method for Process Improvement (SCAMPI [SM]) Methodology.
4. Federal Acquisition Regulation (FAR).

Additional References and Resources

"Assessment and Adaptation," MITRE Project Leadership Handbook, The MITRE Corporation.

Clapp, J. A., and P. G. Funch, March 5, 2003, A Guide to Conducting Independent Technical Assessments, MITRE Center for Air Force C2 Systems.

"Customer and Contractor Interaction," MITRE Project Leadership Handbook, The MITRE Corporation.

"FFRDC Role and Public Interest," MITRE Project Leadership Handbook, The MITRE Corporation.

The Project Management Institute, 2008, *A Guide to the Project Management Body of Knowledge*, (PMBOK Guide), 4th Ed.

Definition: *An independent assessment is a tool that can be used at any point in a program life cycle to provide insight into progress and risks.*

Keywords: *audit, baseline assessment, independent expert review, independent technical assessment, red team, SCAMPI appraisal, Tiger Team*

MITRE FFRDC INDEPENDENT ASSESSMENTS

Planning and Managing Independent Assessments

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to plan, lead, or be team members or subject matter experts of independent review teams.

Introduction

Individual skills and experience are a solid foundation for participation in independent reviews, but completing the review on schedule with quality findings also depends on a disciplined process. This article describes the three phases of a formal review, the essential activities within each phase, why each activity is essential, the risk assumed when an activity is not performed, and lessons learned from actual appraisals and independent reviews.

An independent assessment is a team activity in which the team leader and team members are not members of the organization being assessed. This allows the team to more readily fulfill its charter objectively and without conflicts of interest. The methodology described in this article is based on established appraisal and assessment methodologies and is tailorable to most types of independent assessments.

An independent review can be planned and managed as a project with a three-phase life cycle:

Planning and preparing for an independent review. Paraphrasing Yogi Berra, 90 percent of a review is in the planning; the other 50 percent is executing the plan and delivering the findings. In this phase, we emphasize the critical importance of working with the sponsor to identify and document all relevant details of the review from start to finish. The overarching product of this phase is a review plan, a contract signed by sponsor and the team leader.

Executing the plan. In this phase, the review team interacts more broadly with the organization under review and executes the review plan as approved by the sponsor.

Preparing and delivering final findings. This section describes how the review team develops and presents defensible findings from evidence they have collected, or from their analyses.

Planning and Preparation

> Give me six hours to chop down a tree, and I will spend the first four sharpening the axe. —*Abraham Lincoln*

The first phase of an independent review is the most important. The sponsor (whoever is paying for the review) and review team leader meet face-to-face, and develop a review plan that includes a charter for the review team, a clear specification of objectives and issues to be resolved, a schedule and related dependencies, communications requirements, and resource requirements such as funding. After the initial meeting, the team leader prepares a draft plan for the sponsor's review and approval.

As illustrated in the standard independent review life cycle in Figure 1, reviews must be planned and managed as projects. That is, each should have a beginning and an end, sufficient resources, and a schedule that divides the review into phases with entry and exit criteria. During the review, the team leader is responsible for monitoring and controlling the review against the plan and for all communications with relevant stakeholders.

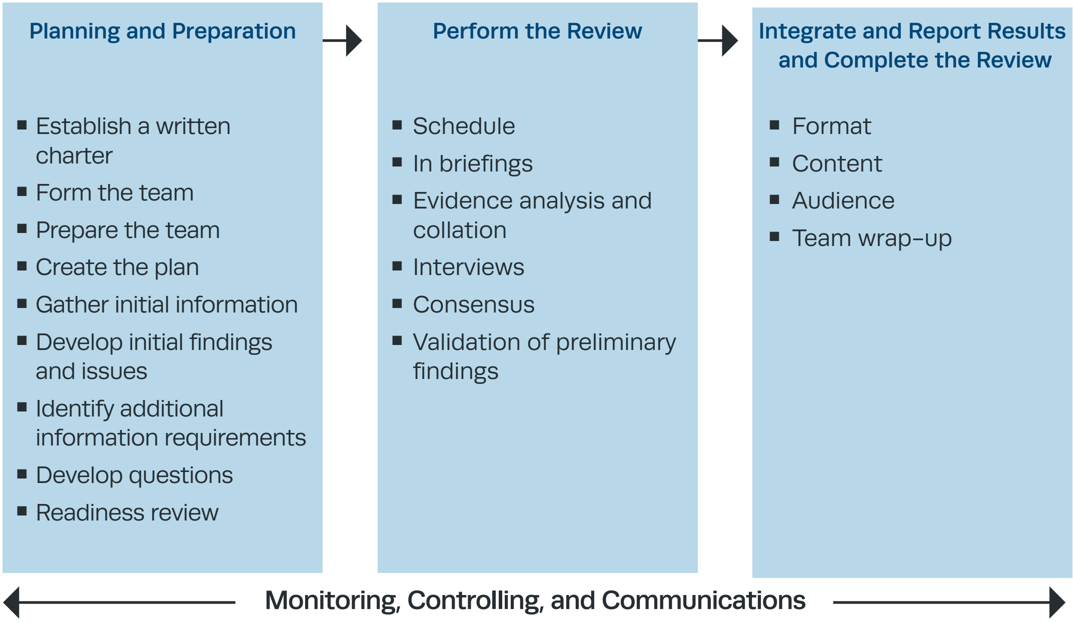
The team leader is also responsible for:

- Re-planning as needed
- Team selection, team-building, and team performance
- Resolution of conflicts and impasses (must be able to motivate, adjudicate, and cajole)
- Reporting results or findings to the sponsor and other stakeholders as required

Planning must be thoroughly documented because every planning element is a potential element of risk. Elements that occasionally get overlooked include names of review participants; the ability of personnel to participate; security and travel requirements; commitment to schedules, particularly during the execution phase and format/content of the final report (PowerPoint? hard copy, Word file, or both?) so the sponsor knows what to expect.

Plan Content. When developing your review plan, include the following sections:

- Cover page (include date and current version numbered) and revision page
- Context information (Organization size and other data, how the review came about)
- Purpose and objectives (Why and what the sponsor expects to gain from the review)



Planning and Preparation

- Establish a written charter
- Form the team
- Prepare the team
- Create the plan
- Gather initial information
- Develop initial findings and issues
- Identify additional information requirements
- Develop questions
- Readiness review

Perform the Review

- Schedule
- In briefings
- Evidence analysis and collation
- Interviews
- Consensus
- Validation of preliminary findings

Integrate and Report Results and Complete the Review

- Format
- Content
- Audience
- Team wrap–up

Monitoring, Controlling, and Communications

Figure 1. Standard Independent Review Life Cycle

- Review team charter (To establish the independent review and the authority of the review team, include a charter signed by the sponsor and the team leader that details preliminary requirements/scope, dependencies, and constraints of the review, and conveys purpose and scope of the review to the subjects of the review)
- Key participants (Sponsor, review team, interviewees, evidence providers, presenters)
- Scope (Sponsor's needs and expectations; what the team will do and deliver. Leave as little as possible to interpretation, and update the plan whenever scope changes.)
- Schedule (Top-level at first, then add details as the review unfolds; "extreme detail" for the execution phase)
- Reference standards or models (Standards, models, etc., used for analysis by the team)
- Dependencies (e.g., among participants, schedule, other activities)
- Constraints (Availability of personnel, time, staffing, funding, tools, facilities, security)
- Resource Requirements (Funding, people, time, facilities, tools)
- Logistics (On-site support, escorts, working hours, Internet access, printers, copiers, phones)
- Risk management (What might go wrong, probability, consequences, mitigation steps)
- Roles/responsibilities (Who does what)
- Training (What may be required to perform the review)
- Security (Facility/information access, on-site escorting, visit requests, and clearances)
- Communications plan (Who talks to whom, when, why, what must be documented, etc.)
- Management reviews (Status report, issues, triggers for ad hoc meetings)
- Deliverables/format (Program or project-specific, tabletop, or formal presentation, level of detail)
- Ownership of the results (Usually the agency or organization that paid for the review)
- Signature page (Sponsor and review team)

Team Selection. After establishing the scope of the review, the team leader selects a team. Candidates must have an appropriate level of engineering and management experience (rookies are not recommended), and they may also need specific technical domain experience. Team members must know how work is done in the context of the project or program being appraised. Is it a Department of Defense (DoD) acquisition program? Homeland Security? Defense Information Systems Agency (DISA)? Internal Revenue Service (IRS)? Find team members who have worked in those arenas. Previous review experience is recommended; however, depending on the appraisal and team size, including one or two inexperienced team members should be con sidered as a means to grow the organization's independent assessment bench strength. It is strongly recommended that candidates be able to commit full-time to the schedule. Part-time team members are a significant risk, so qualified alternates are recommended.

After the team is selected and before the execution phase begins, a full day should be reserved for team building where the team meets, reviews the plan (especially the schedule), develops analysis strategies, and allocates workload. The team may be asked to sign non-attribution statements and non-disclosure agreements, which guarantee that nothing observed, read, or heard during the review will be attributed to a person or project, and that all intellectual property rights of the organization(s) being appraised will be respected.

Initial Findings and Readiness Review. At this point the team should assess the availability of information or evidence needed and make a preliminary feasibility assessment (is there enough time/information/etc., to conduct the review as planned?), then deliver a readiness assessment to the sponsor. Preliminary recommendations can range from "it appears that enough information is available to complete the review—we recommend proceeding with the review as planned" to "we have questions" to "we recommend changing scope" to "we recommend delaying the review—you need more time to prepare."

Table 1. Risks Assumed When Planning Activities Are Not Performed

Activity	Risk
Team leader and sponsor meeting to develop preliminary inputs	Weak basis for further planning: Scope poorly defined Schedule ambiguous No authority to proceed
Establish a written charter	No formal contract between the sponsor and the review team Without authority from the sponsor, the review becomes a lower priority in the organization
Obtain and review initial program information	Too many assumptions Reduced objectivity
Select and build a team	Inappropriate knowledge and skill sets Inconsistent review/analysis methods among sub-teams Team member absenteeism
Develop initial issues	Time lost trying to focus the review at a later stage
Develop the review team's methodology	Inconsistent findings Challenges to findings

Best Practices and Lessons Learned

- Meet with the sponsor, not a delegate.
- Don't start the review without a signed charter and a signed review plan.
- Expect the review to be seen an intrusion or new impediment to progress by the subjects of an independent review. They will, of course, want to be fully engaged in the day-to-day activities of their project. Ask the sponsor to send a copy of the charter to those who will be involved in the review. This will establish the team's authority and its level of access.
- Keep scope as narrow as possible in order to produce supportable and usable findings.
- Activities in scope must be achievable.
- Establish an understanding with the sponsor about the constraints that are placed on the review team and its activities.
- Schedule interviews well in advance. Ask for early notification of cancellations. Be efficient with the time of those being interviewed. They may already be stressed or behind schedule.
- Update the review plan whenever something changes, and publish revisions.
- Review team composition and interpersonal skills of team members are key.
- Team building really pays off.
- Use mini-teams wherever and whenever possible.
- Don't wait until the execution phase to begin planning how the team will locate or identify the evidence they need.
- Start to finish for an assessment should be 30-60 days, depending on scope and team size.

Executing the Plan

During this phase, the appraisal team interacts with the organization and its personnel. The team leader briefs the appraisal plan and the organization presents contextual information about itself.

The team then collects and analyzes evidence by comparing work products, observations (e.g., demonstrations), and oral evidence against the standard or model agreed upon for the review. The team leader monitors team progress, redistributes workload as needed to maintain schedule, and meets daily with the entire team to assess progress. Midway through this phase, the team should conduct a more detailed progress review.

After this internal review the team leader meets with the sponsor, describes issues that warrant attention, and presents recommendations, for example, to expand or reduce the appraisal scope and to continue or terminate the appraisal.

If the sponsor says to continue the appraisal, the team completes the preliminary findings using a consensus decision-making process, which requires any negative vote to be resolved before there is a finding. Preliminary findings should be presented to and validated by organization personnel involved in the appraisal. They alone are allowed to attend the presentation of preliminary findings because findings, if based on incomplete evidence or misinterpreted interview statements, could be incorrect. After the presentation, give the organization personnel a few hours to present new evidence that might change a preliminary finding. Evidence review is then closed, and the team develops its final findings report.

Best Practices and Lessons Learned:

- Have a detailed schedule and stick to it.
- Maintain objectivity and fairness—avoid "witch hunts."
- Find ground truth and evidence to support conclusions.
- Report to the sponsor when independent assessment risks are happening (e.g., participants are not participating).

Table 2. Risks Assumed When Execution Phase Activities Are Not Performed

Activity	Risk
Opening briefs	Lost opportunity to present evidence Confusion about who is supposed to participate in events Timing and location of events Schedule
Detailed schedule	Wasted time Cancellations Scramble to find rooms
Consensus	Diluted findings Customer confusion
Validation of preliminary findings	Quality of final findings Customer satisfaction

Final Findings: Preparation and Delivery

Final findings, the ultimate deliverable of the review, should address all issues and questions identified in the scope statement of the plan. They must be supportable (i.e., developed by the review team from evidence or the results of analyses using a consensus method). The team should review/polish the final findings before delivering them to the sponsor, and then present them as required by the review plan. After the presentation, a record of the appraisal is given to the sponsor and to others authorized by the sponsor. The sponsor alone owns the information presented in the briefing and controls its distribution.

Finally, the independent review team should conduct a wrap-up session to record lessons learned and to discuss possible next steps.

Best Practices and Lessons Learned

- Stay focused until the review is over.
- Tiny details that are missed can spoil several weeks of excellent work.
- The team leader is not necessarily the best presenter for every element of scope.
- Record lessons learned before team members return to their regular jobs.

Table 3. Risks Assumed If Final Phase Activities Are Not Performed

Activity	Risk
Team review of final findings	Items not covered or completely understood Presentation assignments not finalized
Establish delivery method	An uncoordinated presentation “Winging it” in front of a senior audience Best person not presenting a finding
Coordinate the final findings brief	Obvious things not covered: Time of presentation not advertised Poor availability of attendees Room reservation not made No audio-visual setup
Team wrap-up	Lessons learned not tabulated No coordination of potential next steps

References and Resources

“Assessment and Adaptation,” MITRE Project Leadership Handbook, The MITRE Corporation.

Clapp, J.A., and P.G. Funch, March 5, 2003, A Guide to Conducting Independent Technical Assessments, MITRE Center for Air Force C2 Systems.

SEPO Program Assessment Toolkit, viewed February 4, 2010.

Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) Methodology.

The Project Management Institute, 2008, *A Guide to the Project Management Body of Knowledge* (PMBoK Guide), 4th Ed.

SE Life-Cycle Building Blocks

MITRE

Introduction

MITRE systems engineers (SEs) orchestrate the complete development of a system, from a need through operations to retirement, by applying a set of life-cycle building blocks. SEs are expected to understand and work with fundamental building blocks for engineering systems, regardless of the specific life-cycle methodology used. They are expected to define systems conceptually, transform user needs into system requirements, and develop and assess architectures. They are expected to compose and assess alternative design and development approaches; develop test and certification strategies; monitor and assess contractor efforts in design, development, integration, and test; and assist with field deployment, operations, and maintenance.

Background

All systems engineering models and processes are organized around the concept of a life cycle. Although the detailed views, implementations, and terminology used to articulate the SE life cycle differ across MITRE's sponsors and customers, they all share fundamental elements.

For example, Department of Defense (DoD) Instruction 5000.02 [1] uses the following phases: materiel solution analysis, technology development, engineering and manufacturing development, production and deployment, and operations and support; however, this conceptualization of the system life cycle is by no means unique.

ISO/IEC 15288 [2] is an international systems engineering standard covering processes and life-cycle stages. It defines a set of processes divided into four categories: technical, project, agreement, and enterprise. Example life-cycle stages described in the document are: concept, development, production, utilization, support, and retirement. The International Council on Systems Engineering (INCOSE) uses a consistent approach in its Systems Engineering Handbook, version 3.1 [3].

A V-model [4] is a common graphical representation of the systems engineering life cycle (Figure 1). The left side of the V represents concept development and the decomposition of requirements into function and physical entities that can be architected, designed, and developed. The right side of the V represents integration of these entities (including appropriate testing to verify that they satisfy the requirements) and their ultimate transition into the field, where they are operated and maintained. The model we use in this guide is based on this representation. For each phase, we have written articles that succinctly describe the major activities in each cycle. They are summarized in below.

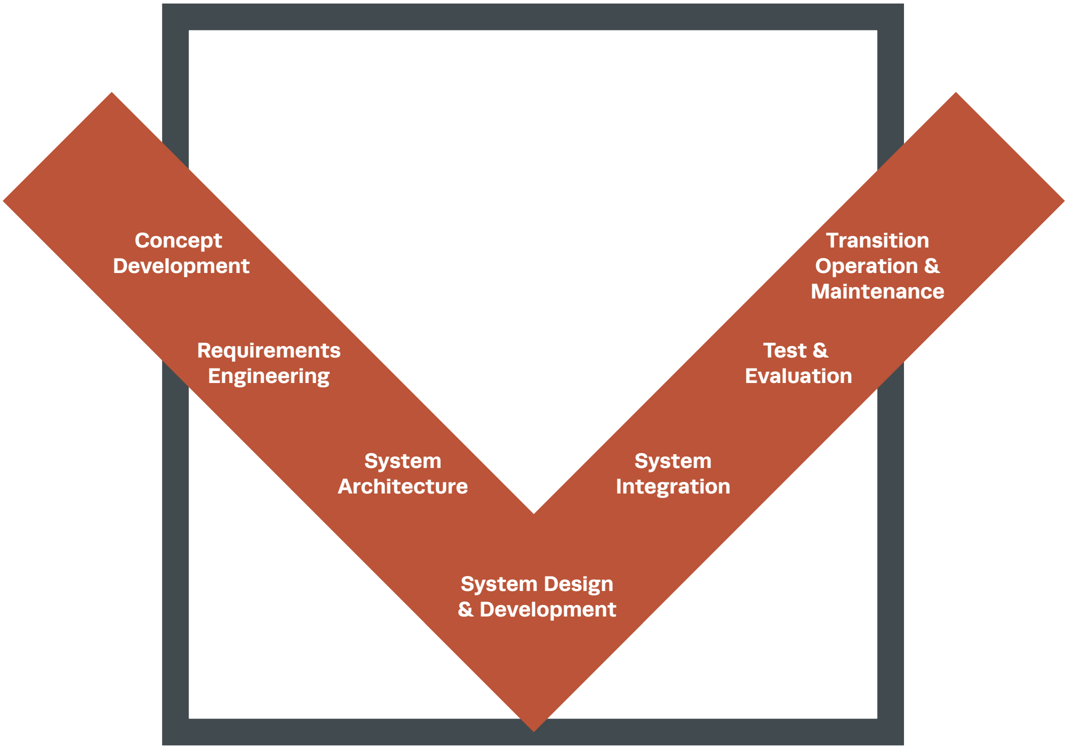


Figure 1. V-model

Concept Development

This first phase is concerned with transforming a user's expression of an operational need into a well-defined concept of operations, a high-level conceptual definition, and a set of initial operational requirements. Articles in this topic area include "Operational Needs Assessment," "Concept of Operations," "Operational Requirements," and "High-Level Conceptual Definition."

Requirements Engineering

In this phase, detailed system requirements are elicited from the user and other stakeholders, the requirements are further analyzed and refined, and plans and processes for managing the requirements throughout the rest of the system life cycle are developed. With today's complex systems, there is always a degree of instability and uncertainty with the requirements, so methods to accommodate this are included as well during this phase. Articles in this topic area include “Eliciting, Collecting, and Developing Requirements,” “Analyzing and Defining Requirements,” and “Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation.”

System Architecture

Once the requirements are expressed and folded into a management process, a system architecture can be described. The architecture will be the foundation for further development, integration, testing, operation, interfacing, and improvement of the system as time goes on. In the system architecture articles, we discuss various architecture patterns (e.g., service-oriented architecture), architectural frameworks (e.g., DoDAF [architectural framework]), and formal processes for developing architectures. Articles in this topic area include “Architectural Frameworks, Models, and Views,” “Approaches to Architecture Development,” and “Architectural Patterns.”

System Design and Development

At this point in the system life cycle, a complete and comprehensive description of what and how the system is expected to perform has been developed along with an architectural representation to guide the actual design and development of the hardware, software, and interfaces. Articles in this topic area include “Develop System-Level Technical Requirements,” “Develop Top-Level System Design,” and “Assess the Design’s Ability to Meet the System Requirements.”

Systems Integration

During the design and development phase, all of the system’s subsystems are complete. In this next system integration phase, the system’s components and its interfaces with other systems are integrated into an operational whole. Articles in this topic area include “Identify and Assess Integration and Interoperability (I&I) Challenges,” “Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies,” “Assess Integration Testing Approaches,” and “Interface Management.”

Test and Evaluation

Because the system is completely designed at this point, it is now necessary to test the system to see if it fulfills the users’ needs (verification) and all of the defined requirements (validation). Testing at this phase also involves properties such as reliability, security, and interoperability. Articles in this topic area include “Create and Assess Test and Evaluation Strategies,” "Assess Test and Evaluation Plans and Procedures," and "Create and Assess Certification and Accreditation Strategies."

Implementation, Operations and Maintenance, and Transition

Finally, to ensure a successful transition of the system into the field, plans and procedures must be developed for operations and maintenance. Because the technological underpinnings of a system are constantly changing, product improvements—including the insertion of new technologies—must be planned for.

Other SE Life–Cycle Building Blocks Articles

This topic is a staging area for articles on subjects of relevance to SE Life-Cycle Building Blocks but that don't neatly fit under one of its other topics. In most cases, this is because the subject matter is at the edge of our understanding of systems engineering, represents some of the most difficult problems MITRE SEs work on, and has not yet formed a sufficient critical mass to constitute a separate topic.

The system life cycle just described is rarely, if ever, as linear as this discussion might imply. There are often iterative cycles, missing phases, overlapping elements, etc. Additionally, processes and activities may apply to more than one phase in a system life cycle, which are better envisioned as threading through or overarching the other building blocks. Articles in this topic area include "Spanning the Operational Space—How to Select Use Cases and Mission Threads," "Acquiring and Incorporating Post-Fielding Operational Feedback into Future Developments: The Post-Implementation Review, Test and Evaluation of Systems of Systems," and two articles on modeling and simulation —"Verification and Validation of Simulation Models" and "Affordability, Efficiency, and Effectiveness (AEE)."

References and Resources

1. Department of Defense, December 8, 2008, "Operation of the Defense Acquisition System," Instruction Number 5000.02.
2. ISO/IEC 15288, 2002, Systems Engineering—System Life Cycle Processes.
3. International Council on Systems Engineering (INCOSE), January 2010, INCOSE Systems Engineering Handbook, Ver. 3.2, INCOSE-TP-2003-002-03.2.
4. Wikipedia contributors, "V-Model," Wikipedia, accessed January 13, 2010.

SE Life-Cycle Building Blocks Contents

Concept Development

Definition: *Concept development is a set of activities that are carried out early in the systems engineering life cycle to collect and prioritize operational needs and challenges, develop alternative concepts to meet the needs, and select a preferred one as the basis for subsequent system or capability development and implementation.*

Keywords: *analysis, concept, definition, development, exploration, requirements, systems engineering*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to develop and agree upon a working description and view of how the systems or capabilities to be developed will be used, how they will function in their expected environments, what top-level requirements they will satisfy, and their high-level conceptual design. MITRE SEs are expected to be able to use a variety of approaches to elicit user needs and explore and assess alternative concepts to meet them, including prototypes (see the article "Competitive Prototyping" in the SEG's Acquisition Systems Engineering section) and experiments that involve users, developers, and integrators.

Context

Concept development takes place early in the systems engineering life cycle. The success of the subsequent development of a system or capability can be critically dependent on the soundness of the foundation that is laid during the concept development stage. In their definitions of concept development, Kossiakoff and Sweet [1] highlight phases of needs analysis (valid need and practical approach), concept exploration (performance to meet the need, feasible cost-effective approach), and concept definition (key characteristics that balance capability, operational life, and cost).

In this guide, concept development is described as four activities that identify and characterize user needs:

1. **Operational Needs Assessment:** The application of operational experience to identify and characterize gaps in existing capabilities that are significant impediments to achieving the mission-area objectives.
2. **Concept of Operations:** A description of a proposed system's characteristics in terms of the needs it will fulfill from a user's perspective.
3. **Operational Requirements:** Statements that formally, unambiguously, and as completely as possible, identify the essential capabilities and associated performance measures.
4. **High-Level Conceptual Definition:** A clear description or model of the characteristics or attributes needed to address a specific set of requirements or capabilities.

MITRE systems engineers (SEs) should understand that, like the environment, operational needs and requirements cannot be viewed as static. User needs change, their priorities change, and the technology to enable them changes. This means that requirements cannot be viewed as cast in stone with subsequent systems engineering aligned to an inflexible baseline. Trade-off analyses may be required more or less continuously to ensure effective capabilities are delivered to meet users' immediate and evolving needs.

Many processes, methods, and tools are available for conducting concept development. Of critical importance are the initial questions that must be answered early to get the requirements elicitation done right. These questions apply whether developing a product (system, capability, or service) or an operational structure to employ the product. MITRE SEs must ask more than "who, what, where, when, why, and how." They must develop specific questions to address broader issues, such as:

- What are the current deficiencies and gaps?
- What are the external constraints?
- What are the real-world performance drivers?
- What are the operational, security, and support concepts?
- Is it feasible technically, economically, and in a timely manner?
- What are the associated, external, and interfacing activities?
- What happens if any of the questions above cannot be answered?

The insight obtained from such questions will likely lead to a preferred concept to satisfy the end users' needs and provide a sound basis for development. MITRE SEs should seek out guidance on customer-specific elements and approaches (i.e., Air Force [AF] Concept Development) [2].

Best Practices and Lessons Learned

Questions to Ask (adapted from [1]):

To meet the need, have at least two alternative concepts been developed and evaluated? The purpose of alternatives is to stimulate thinking to find simpler, faster, or cheaper solutions.

What technologies does each concept depend on? Have they been critically assessed for maturity? Are there more mature technologies that can support the concepts? A number of high-level government studies concluded that the development of risky new technology to support a major acquisition program is a leading contributor to cost and schedule overruns and failure to deliver. Increasingly, government departments and agencies are requiring mature technologies before allowing an acquisition program to go on contract.

Is the proposed solution right-sized economically? Would delivering 80 percent of the solution delivered early be of greater value to accomplishing mission success? Beware attempts to satisfy that last, lone requirement before getting capabilities to the users.

Have external interface concepts, requirements, and complexities, including dependencies on other programs, been identified and addressed? Are there one or more specific "in-hand" alternatives that will enable the concept to be negotiated or realized, particularly when the concept relies on capabilities to be delivered by third-party programs over which your program has no control or little influence? Complex, ill-defined external requirements and interfaces can be a major source of requirements instability during the development phase. This can be important when a system must operate in a system-of-systems environment.

Are concepts and requirements firmly tied to operational and mission success versus individual user/organization preferences? Achieving capabilities or demonstrating critical subsystems that transcend individual perspectives while meeting operational timelines is important for achieving service quickly and cost effectively, and to begin the process of incremental improvements based on operational experience, needs, and capability evolutions.

References and Resources

1. Kossiakoff, A., and W. Sweet, 2003, *Systems Engineering: Principles and Practice*, John Wiley & Sons, Inc.
2. Air Force Policy Directive 10-28, September 15, 2003, AF Concept Development.

Additional References and Resources

Air Force Studies Board Division on Engineering and Physical Sciences, 2008, Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition.

International Council on Systems Engineering website.

International Organization for Standardization, 2008, "ISO/IEC Standard 15288, Systems and Software Engineering—System Life Cycle Processes."

Rechtin, E., 1991, *Systems Architecting: Creating and Building Complex Systems*, Prentice Hall.

Stevens, R., P. Brook, K. Jackson, and S. Arnold, 1998, *Systems Engineering: Coping with Complexity,* Prentice Hall.

The MITRE Corporation, "Concept Definition," MITRE Systems Engineering Competency Model, section 2.1, accessed February 22, 2010.

Wikipedia contributors, "Systems Engineering," accessed February 22, 2010.

Definition: *An operational needs assessment identifies and characterizes gaps in existing capabilities that are significant impediments to achieving the mission-area objectives. It does so through the application of operational experience in a particular mission/business area, knowledge of related processes and elements involved in conduct of the mission, and knowledge of the mission's objectives and measures of success/effectiveness.*

Keywords: *acquisition development program, capability-based assessment, CBA, operational needs assessment, program management*

CONCEPT DEVELOPMENT

Operational Needs Assessment

MITRE SE Roles and Expectations: The operational or capability-based needs assessment is typically the responsibility of the operational requirements organization of the system's or capability's end user. MITRE systems engineers (SEs) are often requested to support such assessments, or even develop complete assessments for review and approval. Key roles for MITRE SEs in this process may include ensuring that operational needs statements are clear and complete; understanding and conveying areas of uncertainty or flexibility; ensuring analyses contain appropriate attributes and associated metrics supported by analytical or operational evidence, and are clearly tied to operational goals; modeling/prototyping/experimenting on needs/gaps for clarity, feasibility, and integration; assessing technical readiness/risk/feasibility of technology-driven capability needs; and identifying risk/cost drivers in capability needs.

MITRE SEs should have a sound understanding of the principles of needs assessment, the needs assessment process of the supported government element, and the political, business, operational, and technical (enterprise) context of the capability area of interest, as well as how the government customer/sponsor intends to continue evolution and sustainment of the product following product delivery.

Background

Operational needs assessments are frequently the initial step toward a new development or process improvement program. A needs assessment is conducted by the user community to determine the best capabilities that will help users accomplish their operational tasks. These assessments are accomplished through operational experiments, exercises, modeling and simulation of user tasks/operations, etc. From these, users formulate their needs and requirements. Assessment products are the basis for applicable technology assessments, solution alternatives analyses, operational requirements definitions, and ultimately the acquisition program (or programs). An operational needs assessment defines the business and mission need for providing systems, services, capabilities, or platforms to end users and other stakeholders, and develops a business case that justifies the return on investment in order to obtain funding for a system or multiple systems [1, 2].

New needs can arise for a number of reasons: new goals (e.g., manned mission to Mars), new conditions (new or improved threats, deterioration/discontinuation of existing capabilities, hardware/software end of life), changing processes/regulations (new laws, new organizational responsibilities, new relationships), introduction of new technologies that enable previously unattainable capabilities or enable improved or more cost-effective solutions to existing capabilities, etc.

Why do we perform operational needs assessments? First, we are typically required to do so by Department of Defense (DoD), Federal Aviation Administration, Internal Revenue Service, and other formal organizationally run programs. But even if not required, an assessment of operational needs and lessons learned provides greater understanding by the user, acquisition, development, and support communities so that needs can be satisfied with capabilities, products, or improved processes valuable to mission success. The government has limited resources to obtain its goals. Operational needs must be described and quantified in the context of operational user needs and goals for decision makers to assess their validity, importance, and urgency in the context of competing needs, and determine the risk of not obtaining the capability. If new needs can be most cost-effectively met by changes to DOTLPF (Doctrine, Organization, Training, Logistics, Personnel, Facilities), new materiel solutions may not be necessary. In any case, the operational needs must be defined and quantified in a manner that allows for assessment of the most cost-effective solution alternatives for the need.

Process

In the context of a systems engineering life cycle, an operational needs assessment forms the basis for defining requirements for a program and a system. It occurs as an initial step in the life cycle but also must be continuously addressed as operational environments, evolutionary strategies, priorities, and funding change. As part of initial program planning, MITRE is frequently involved in the establishment of systems engineering processes, of which operational needs assessment is an important one. The process elements are:

- Determine the specific requirements of the needs assessment process that apply.
- Identify specific stakeholders in this particular needs assessment process, including their responsibility, goals, and roles/relationships.
- Identify and obtain support from operational or capability domain experts.
- Develop a needs assessment plan and schedule, which can include scenario-driven experiments, gap analysis, trade-off studies, etc.
- Identify and put in place any analytical tools necessary to define and quantify needs.
- Implement and conduct needs assessment.

Operational Needs Considerations

As an example, the DoD Joint Capability Integration and Development System (JCIDS) process [3, 4] includes several steps leading to an operational requirements document (capability development document [CDD]) for acquisition of a system. Although other government departments and agencies may have different specifics, the basic approach has general applicability. It begins with a capabilities-based assessment that identifies the mission, the capabilities required, and their associated operational characteristics and attributes, capability gaps, potential solutions (e.g., processes, systems, technologies), and associated operational risks. If a DOTLPF assessment determines that a new system (materiel) capability is required, an initial capability document (ICD) is developed. The ICD contains the definition of the capabilities needed along with their important attributes and associated metrics. This is used as the basis for an analysis of alternatives (AoA), which provides quantitative cost/effectiveness trades for alternative approaches to providing the capability. The results of this AoA are then used to develop the solution-approach specific CDD.

The ICD is the repository of the capability needs assessment results. The needs statements should have the following attributes:

- **Enterprise and Operational Context:** It is important that needs be considered in an enterprise context. If related enterprise capabilities can address part of the need, define the unique characteristics of the new need in this context.
- **Complete (End-to-End) Need Defined:** Ensure that the need is defined as completely as possible (e.g., detect, identify, and defeat incoming cruise missiles vs. detect incoming cruise missiles). Recognize where areas of uncertainty remain or areas of flexibility exist.
- **Conditions/Scenario:** Define the conditions/scenario under which the capability/need will exist (e.g., indications and warning vs. major combat operations, jamming vs. clear, communications/power outage).
- **Attributes/Metrics:** Consider quantifiable metrics for the capability that define how much, how well, how often, and how quickly the capability must perform. These metrics should be directly related to mission goals. Again, recognize where areas of uncertainty remain or flexibility exist.
- **Growth/Extensibility:** If current needs are expected to increase or expand in the future, state those expectations so that expandability/extendibility of solution approaches can be properly taken into account and hooks are put in place to enable those extensions. Note that making design choices that favor enhanced adaptability is always prudent.
- **Independent of Solution Approach:** If needs are stated as a particular solution approach, they can eliminate consideration of more effective approaches to meeting the actual need. Thus, needs should be articulated in terms that describe a successful operation or mission instead of a proposed solution.

Lessons Learned

Beware solutions masquerading as needs. Operational or capability needs are often represented by users in terms of specific solution approaches. This can result from marketing or technology demonstrations, familiarity with a specific solution approach, or preference for a specific solution/approach due to unstated (or unrecognized) aspects of the need (political, economic, etc.). The challenge is to extract from the users the full definition of the underlying capability needed, and obtain stakeholder concurrence that the focus needs to be on those identified capabilities, not a solution-based approach. The best approach for understanding the needs is by observing and talking with the actual end users. It may be a challenge to get their time and access. If so, consider a user "surrogate," including MITRE employees with recent operational experience.

State needs unambiguously. Key attributes and metrics are frequently missing, stated in ambiguous terms, or stated with no corroborating analysis or evidence basis. The challenge is to clarify needs in unambiguous terms, with attributes and metrics directly related to mission goals (measures of effectiveness), and supported by analysis or operational evidence.

Get all relevant views. Operational needs can be driven by a subset of the key stakeholders (e.g., system operators vs. supported operational elements), and thereby miss key capability needs. The challenge is to ensure that all key stakeholders' needs are taken into consideration.

One size may (or may not) fit all. The union of a set of needs may lead to a solution that is too cumbersome to implement cost-effectively. Remember that multiple solutions to subsets of needs, or satisfying additional needs by iterative solution augmentations, may sometimes be the most practical approach, assuming operational needs can be adequately met. Methods such as modeling and simulation and prototyping/experimentation allow an examination of the needs-satisfaction approaches and evolution and help plan the augmentations that best satisfy operational needs and missions over time.

The educated consumer is the best customer. Particularly in the case of new technology-driven needs, operational requirements contributors can be unfamiliar with potential capabilities, the user's concept of operations or concept of use, organizational, and political implications. The challenge is to educate users on capabilities, limitations, cost drivers, and operational implications of the new technologies so that the capability delivered provides the best cost/performance balance for the customer. Prototyping and experimentation, particularly with heavy user involvement, can help educate not only the end user, but the SEs as well. For best practices and lessons learned, see the article "Competitive Prototyping" in the SEG's Acquisition Systems Engineering section.

References and Resources

1. The MITRE Corporation, "Concept Development," MITRE Systems Engineering Competency Model, section 2.1, accessed February 22, 2010.
2. An example of a document that contains operational needs is "U.S. Citizenship and Immigration Service Concept of Operations," October 22, 2009, USCIS Transformation Program.
3. Joint Capabilities Integration and Development System (JCIDS), March 1, 2009, Chairman of the Joint Chiefs of Staff Instruction 3170.01.
4. Manual for the Operation of the Joint Capabilities Integration and Development System, updated July 31, 2009.

Definition: *A Concept of Operations (CONOPS) is a user-oriented document that "describes systems characteristics for a proposed system from a user's perspective. A CONOPS also describes the user organization, mission, and objectives from an integrated systems point of view and is used to communicate overall quantitative and qualitative system characteristics to stakeholders [1]."*

Keywords: *concepts, CONOPS, operational concept description, operational concepts, operational scenarios, system concepts, use cases, user needs, user/system roles, viewpoints*

CONCEPT DEVELOPMENT

Concept of Operations

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand and recommend the development and use of a CONOPS as a tool throughout the systems engineering life cycle to communicate user needs and system characteristics to developers, integrators, sponsors, funding decision makers, and other stakeholders. In some cases MITRE SEs may be asked to support the development of a CONOPS.

MITRE SEs should be able to apply systems engineering methods to map user (operational) needs to system requirements, functions, and conceptual system designs. They should also be able to develop test requirements that are traceable to system requirements and user needs. In addition, they should test operational concepts (concept validation) and user utility as described in the CONOPS.

Background

The Office of Management and Budget defines a CONOPS as describing "the proposed system in terms of the user needs it will fulfill, its relationship to existing systems or procedures, and the ways it will be used. CONOPS can be tailored for many purposes, for example, to obtain consensus among the acquirer, developers, supporters, and user agencies on the operational concept of a proposed system. Additionally, a CONOPS may focus on communicating the user's needs to the developer or the developer's ideas to the user and other interested parties [2]."

The purpose of a CONOPS is to describe the operational needs, desires, visions, and expectations of the user without being overly technical or formal. The user, developer, or both may write CONOPS, often with help from MITRE SEs. The CONOPS written by a user representative communicates the overall vision for the operational system to the organizations (e.g., buyer, developer) that have a role in the system acquisition and/or development effort. A CONOPS can also be written by the buyer, developer, or acquirer to communicate their understanding of the user needs and how a system will fulfill them. In both cases, the CONOPS is intended to facilitate a common understanding of ideas, challenges, and issues on possible solution strategies without addressing the technical solution or implementation; it is often a first step for developing system requirements.

As systems continue to evolve in complexity, SEs and mission owners can use a CONOPS to develop and sustain a common vision of the system for all stakeholders over the system's life cycle. The original CONOPS written at the beginning of system acquisition should be updated after developmental and operational testing, to convey how the system being acquired will actually be used. This update is needed since many final systems include some additional capabilities not originally envisioned at program start, and may not include some capabilities that were omitted during trade-off analysis. The CONOPS should include the full range of factors that are needed to support the mission (i.e., doctrine, organization, training, leadership, materiel, personnel, facilities, and resources). Post-fielding life cycle costs often dwarf those of the development effort. Therefore, it is critical that the CONOPS provide sufficient information to determine long-term life cycle needs such as training, sustainment, and support throughout capability fielding and use.

A CONOPS should contain a conceptual view of the system (i.e., a preliminary functional flow block diagram or operational architecture) that illustrates the top-level functional threads in the proposed system or situation. A CONOPS should define any critical, top-level performance requirements or objectives stated either qualitatively or quantitatively (including system rationale for these objectives). The SE should consider the CONOPS as a functional concept definition and rationale from the user and customer perspectives.

Multiple CONOPS guidelines, models, and methodologies are available that can be tailored as needed for particular environments or situations. A MITRE SE should be able to determine which CONOPS format, model, or methodology is appropriate for the specific situation, and if (or how) it should be tailored for that system/environment. Johns Hopkins University's Whiting School of Engineering provides an approach to making this decision based on SE analysis of criteria:

- Program risks
- Customer desires, requirements
- Funding constraints
- Market considerations
- Technology considerations
- Nature of the system to be developed.

Sample Methodology

The Institute of Electrical and Electronics Engineers (IEEE) Standard 1362-1998 (IEEE Std 1362-1998), *IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps),* is an example of a well-developed and commonly used SE CONOPS guideline. Several SE organizations, including the International Council on Systems Engineering (INCOSE), currently use the IEEE CONOPS guidelines, which state:

> This guide does not specify the exact techniques to be used in developing the ConOps document, but it does provide approaches that might be used. Each organization that uses this guide should develop a set of practices and procedures to provide detailed guidance for preparing and updating ConOps documents. These detailed practices and procedures should take into account the environmental, organizational, and political factors that influence application of the guide [1].

CONOPS Objectives

In the situation where the operational user has not developed a CONOPS, MITRE SEs should select or recommend a CONOPS guideline or model, and the objectives for developing a CONOPS. They should also consider any guidelines that have been put in place by the organization. The main objective of a CONOPS is to "communicate with the end user of the system during the early specification stages to assure the operational needs are clearly understood and incorporated into the design decisions for later inclusion in the system and segment specifications [1]."

Regardless of who develops the CONOPS, frequent interaction is needed among the end users, MITRE SEs, acquisition organizations, and development, test, and security stakeholders. It may also be the case that the operational user does not understand or cannot envision how new capabilities will operate in their environment, particularly if it is a new type of system or operation. In these cases, experiments and prototypes can be of value in illuminating these issues. Additional CONOPS objectives include:

- Provide end-to-end traceability between operational needs and captured source requirements.
- Establish a high-level basis for requirements that supports the system over its life cycle.
- Establish a high-level basis for test planning and system-level test requirements.
- Support the generation of operational analysis models (use cases) to test the interfaces.
- Provide the basis for computation of system capacity.
- Validate and discover implicit requirements.

Critical CONOPS Components

When tailoring IEEE Standard 1362-1998 CONOPS for a specific purpose, noncritical components can be deleted or minimized. However, any CONOPS should always include critical components. These components are contained in IEEE Standard 1362-1998 (discussed below):

- The existing system (manual or automated) the user wants to replace.
- Justification for a new or modified system (including restrictions on that system).
- A description of the proposed system.
- Scenarios highlighting use of the system in the user's environment, including internal and external factors.

For a software-intensive capability, the CONOPS might have a greater emphasis on the information system perspective of the users' needs and developers' products, concentrating on software feasibility and software requirements.

Systems Engineering Applications for a CONOPS

MITRE SEs should be able to use various iterations of a CONOPS as a tool throughout the systems engineering life cycle to communicate user needs and system characteristics to developers, integrators, sponsors, funding decision makers, and stakeholders. IEEE Standard 1362-1998 guidance on the application of a CONOPS provides additional clarification. "The ConOps approach provides an analysis activity and a document that bridges the gap between the user's needs and visions and the developer's technical specifications." The ConOps document also provides the following information:

- A means of describing a user's operational needs without becoming bogged down in detailed technical issues that shall be addressed during the systems analysis activity.

- A mechanism for documenting a system's characteristics and the user's operational needs in a manner that can be verified by the user without requiring any technical knowledge beyond that required to perform normal job functions.
- A place for users to state their desires, visions, and expectations without requiring the provision of quantified, testable specifications. For example, the users could express their need for a "highly reliable" system, and their reasons for that need, without having to produce a testable reliability requirement. [In this case, the user's need for "high reliability" might be stated in quantitative terms by the buyer prior to issuing a request for proposal (RFP), or it might be quantified by the developer during requirements analysis. In any case, it is the job of the buyer and/or the developer to quantify users' needs.]
- A mechanism for users and buyer(s) to express thoughts and concerns on possible solution strategies. In some cases, design constraints dictate particular approaches. In other cases, there may be a variety of acceptable solution strategies. The CONOPS document allows users and buyer(s) to record design constraints, the rationale for those constraints, and to indicate the range of acceptable solution strategies [1].

Best Practices and Lessons Learned

User's perspective. Use tools and/or techniques that best describe the proposed system from the users' perspective and how it should operate.

Simple and clear. Describe the system simply and clearly so that all intended readers can fully understand it.

User's language. Write the CONOPS in the user's language. Avoid technical jargon. If user jargon is employed, provide a glossary that translates it for nonusers.

Graphics. Use graphics and pictorial tools as much as possible because a CONOPS should be understandable to different types of stakeholders. (Useful graphical tools include, but are not limited to, node-to-node charts, use cases, sequence or activity charts, functional flow block diagrams, structure charts, allocation charts, data flow diagrams, object diagrams, storyboards, and entity relationship diagrams.)

Operational environment. Describe the operational environment in detail to give the readers an understanding of the assumptions, constraints, numbers, versions, capacity, etc., of the operational capability to be used.

Physical environment, safety, security, and privacy. Describe those aspects of the physical environment, safety, security, and privacy that exert influence on the operation or operational environment of the proposed system.

Voluminous descriptions. Include voluminous descriptions, such as a data dictionary, in an appendix, or incorporate them by reference.

References and Resources

1. IEEE Computer Society, March 19, 1998, *IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps)* (IEEE Std 1362-1998).
2. Office of Management and Budget, December 5, 1994, Operational Concept Description (OCD), Data Item Description DI-IPSC-81430.

Additional References and Resources

Fairley, R. E., R. H. Thayer, and P. Bjorke, April 18–22, 1994, *Proceedings of the First International Conference on Requirements Engineering*, pp. 40–47.

CONCEPT DEVELOPMENT

Operational Requirements

Definition: *Operational requirements, the basis for system requirements, "identify the essential capabilities, associated requirements, performance measures, and the process or series of actions to be taken in effecting the results that are desired in order to address mission area deficiencies, evolving applications or threats, emerging technologies, or system cost improvements [1]." The operational requirements assessment starts with the Concept of Operations (CONOPS) and goes to a greater level of detail.*

Keywords: *concept definition, concept development, operational requirements, requirements attributes, stakeholders, user needs, user requirements, users*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to understand the users' needs based on the operational needs assessment (i.e., what mission-area capability gaps need to be addressed). They must be able to analyze the needs identified by the capability gaps and develop or assist in defining the operational and top-level characteristics or requirements of the system. They also should use the concept of operations (CONOPS) to understand the operational needs, desires, visions, expectations, performance requirements, and challenges of the system. MITRE SEs, together with the users, developers, and integrators, assist in defining the system operational requirements, ensuring the requirements map to the operational needs assessment and CONOPS. They work closely with the users to define and develop operational requirements that are reasonable and testable.

MITRE SEs are expected to be able to lay out an evolutionary strategy for the requirements that identifies and prioritizes initial capabilities and subsequent capability increments to be implemented over time. This approach allows for rapid delivery of initial capabilities and enables agility in delivering future capabilities that are responsive to changes in the operational environment. MITRE SEs are responsible for identifying and assessing conditions, constraints, conflicting requirements, and organizational issues, including safety and security factors, and reaching a resolution. They will typically work to gain user agreement on the operational requirements, including refining and changing requirements, throughout the system development process. For more information on CONOPS, see the SEG's Concept Development topic.

Background

A key process in the concept development phase is analysis to define the operational requirements of the system. Operational requirements are typically prepared by a team of users, user representatives, developers, integrators, and MITRE SEs and are based on the identified user need or capability gaps (see the article "Operational Needs Assessment"). Establishing operational requirements forms the basis for subsequent system requirements and system design in the system design and development phase.

The operational requirements focus on how the system will be operated by the users, including interfaces and interoperability with other systems. The requirements establish how well and under what conditions the system must perform. The operational requirements should answer:

- *Who* is asking for this requirement? *Who* needs the requirements? *Who* will be operating the system?
- *What* functions/capabilities must the system perform? *What* decisions will be made with the system? *What* data/information is needed by the system? *What* are the performance needs that must be met? *What* are the constraints?
- *Where* will the system be used?
- *When* will the system be required to perform its intended function and for how long?
- *How* will the system accomplish its objective? *How* will the requirements be verified?

Process

The operational requirement definition process includes the following activities:

- Identify stakeholders who will or should have an interest in the system throughout its entire life cycle.

- Elicit requirements for what the system must accomplish and how well. Doing this in the form of operational scenarios and/or use cases can be particularly helpful in discussions with end users.
- Define constraints imposed by agreements or interfaces with legacy or co-evolving enabling systems.
- Establish critical and desired user performance: thresholds and objectives for operational performance parameters that are critical for system success and those that are desired but may be subject to compromise in order to meet the critical parameters. To assess the feasibility of meeting performance, consider, suggest (if appropriate), and help formulate prototypes and experiments to determine whether near-term capabilities can satisfy users' operational performance needs. Results of the prototypes can help determine an evolutionary strategy to meet critical performance. Additionally, technology assessments can help gauge when desired performance might be met in the future.
- Establish measures of effectiveness and suitability: measures that reflect overall customer/user satisfaction (e.g., performance, safety, reliability, availability, maintainability, and workload requirements) [2]. Many of these measures will be used for the test and evaluation life-cycle building phase.

This process is consistent with the standard process for determining any level of requirements. See the SEG's Requirements Engineering topic for a discussion on eliciting, analyzing, defining, and managing requirements and discussions on the characteristics of "good" requirements (e.g., concise, necessary, attainable, verifiable, traceable, implementation free, evolvable). It is also important to establish a requirements baseline that is kept under configuration control (see the SEG's Configuration Management topic). Together with the rationale, this provides an established and complete audit trail of decisions and changes that were made. The configuration baseline will also identify and manage the trade-offs of satisfying near-term requirements versus allocating requirements over the evolution of a system.

Challenges

As you work to determine user needs, capabilities, and requirements, there are likely to be challenges and complications, including:

- It's not clear who the user is.
- Needs are not well stated or understood by the user or customer and therefore not understood by the developer and integrator.
- What is stated may not be what is really needed.
- Needs are too detailed and focus on a solution.
- Implicit or unreasonable expectations may not be achievable.
- Customer or user changes occur during the system development process.

- Needs often evolve or change. Sometimes this is necessary, but "requirements creep" should always be critically assessed. Does it contribute to an immediate, important need? Is it technically feasible? Is it likely to work in the targeted operational environment?
- The concept may not solve the problem.
- Users don't know about current technology.

MITRE SEs should work to overcome these challenges by getting close to the users to understand their needs and environment; help them understand the realm of the possible with current technical capabilities; and create demonstrations for the users illustrating what is possible to meet their immediate and future needs.

Documentation

The operational requirements are captured in a document, model, or specification (e.g., user requirements document, operational requirements document, or capabilities development document). The type of document is dependent on the type of acquisition and the customer organization, (e.g., Department of Defense, Federal Aviation Administration, Department of Homeland Security, Internal Revenue Service, or other government agency). Whatever name and form these documents take, they provide a basic framework for the articulation and documentation of operational requirements to be used by all stakeholders. The complexity of the intended system and its operational context will govern the required level of detail in the operational requirements document. Examples and formats of these documents are found in the references.

Best Practices and Lessons Learned

The following tips from active systems engineering practitioners may help your work through the concept development phase and the operational requirements development process:

Work with the end users early and often. Be sure to fully understand their mission, operational domain, and most important, their constraints. It is helpful to talk their "language." This allows for an easier exchange of ideas, resolution of conflicts, etc. Participate in training and exercises that the user community is involved in to get a firsthand perspective of the operational environment.

Create mutually beneficial interactions. Determine the users' needs by using mutually beneficial studies or analyses, including modeling and simulation, prototypes, and demonstrations where appropriate. These help the users justify and defend capability needs while providing the acquisition organization with requirements and CONOPS to start system development, testing, and fielding.

Organize your thinking before engaging users. It is often difficult for users to develop requirements from scratch. Draft your understanding of their requirements prior to engaging with them and create a straw man for discussion. This provides a good starting point for discussion. They will tell you if it is wrong. Demonstrations of your understanding using executable models or prototypes of capabilities will help both you and the users to engage on the operational needs and realm of the possible.

Help users understand new technology. Provide users with suggestions on how they might employ a new technology. Often, users cannot see past how they do business today. Introducing them to technology might help them break loose from their thought processes, develop new processes, and possibly rethink or refine some requirements. Consider the use of prototypes to help demonstrate possibilities and show users the technical aspects of a potential solution as they identify their operational needs and consider gives-and-takes based on solution feasibility and constraints.

Explain technology limitations clearly and simply. Clearly and simply explain the limitations of the technology to users, including maturity and associated risk. This helps ensure that requirements are achievable, secures their buy-in on what is possible, and stimulates them to think about how to use what they can get. Again, consider using prototypes or experiments of capabilities that can help bring technology issues to the forefront with users.

Engage users throughout the process. It is important to stay engaged with the user community through the system development process. Break down barriers and overcome incorrect and bad perceptions. Ensure that users are involved in the decision making process. Ensure that they are involved in subsequent decision making that concerns trade-offs affecting operational utility or performance. Keep users apprised of schedule and capability impacts. This builds trust and cooperation, facilitates quick turn times on questions, and helps ensure that the users' needs and objectives are met.

Make user satisfaction a priority. Make customer/user satisfaction a key metric for your program.

Make delivery to the users a primary driver. Getting capabilities to users early and often is the best strategy. The users have a mission to satisfy, and every capability that can help them is needed as soon as feasible. Evolve the deliveries over time based on priorities, associated capability, and feasibility of implementation.

Build a foundation of trust. The greatest likelihood of making good decisions occurs when the users and acquisition communities trust each other.

Summary

The following summary points can help the development of operational requirements:

- Requirements define problems while specifications define solutions.
- Make sure your operational requirements are product, service, and solution agnostic (i.e., they do not assume or target a certain solution).

- Make the solution space broad.
- Keep it simple; make it easy for a reader to understand the problem and requirements that address it [3].

Project success is rooted in understanding operational requirements. This requires the user and acquisition communities and other stakeholders to invest the time and effort both early in the concept development process and throughout the development cycle. Skillfully done, this should result in a greater likelihood of fielding a capable initial system and subsequent evolutions that meet user needs within schedule and cost.

References and Resources

1. Kossiakoff, A., and N. Sweet, 2003, *Systems Engineering Principles and Practices*, Wiley & Sons.
2. International Council on Systems Engineering (INCOSE), January 2010, *INCOSE Systems Engineering Handbook*, Version 3.2, INCOSE-TP-2003-002-03.2, p. 58.
3. Celluci, T., November 2008, *Developing Operational Requirements: A Guide to the Cost-Effective and Efficient Communication of Needs*, version 2.0, Department of Homeland Security.

Additional References and Resources

Bahil, T., and F. Dean, 1997, “The Requirements Discovery Process,” SAND—96-2901C, Sandia National Laboratories, Albuquerque, NM.

ESC/EN Requirements Process Toolkit, ENwebWiki, The MITRE Corporation.

“Requirements Management,” ENwebWiki, The MITRE Corporation.

The SEPO Requirements Process Toolkit, The MITRE Corporation.

Definition: *High-level conceptual definition (HLCD) is the explicit construction of the ideas or concepts needed to understand what a system, product, or component is, what it does, and how it is best used. An HLCD is used by the operational users or, more generally, the stakeholder community. The HLCD may also address what a product is not, what it doesn't do, and how it is not well used. The HLCD reflects a shared point of view, conveying a clear description or model of the characteristic or attributes needed to address a specific set of requirements or capabilities.*

Keywords: *acquisition program, concept definition, concept development, early systems engineering*

CONCEPT DEVELOPMENT

High-Level Conceptual Definition

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to develop or help develop a high-level conceptual definition during the concept development phase of system development. They are expected to assess the full breadth of the solution space (trade space) and to consider, refine, discard, or adopt alternative concepts. This assessment is useful as the life cycle continues into acquisition and development. It is a key input to performing the analysis of alternatives to support the acquisition approach (see the article "Performing Analyses of Alternatives"). MITRE SEs are also expected to take operational requirements and translate them into a concept that provides the stakeholder community with a clear and unambiguous definition of the capability, system, product, or component. They are expected to use this concept definition to guide users in refining requirements, reconsidering the concept of operations or employment, and exploring fundamental implementation approaches. Techniques such as prototyping and experimentation with user community involvement can help highlight aspects of the trade space and illuminate alternatives for addressing user operational concepts and needs.

In developing an HLCD, MITRE SEs are expected to create a user-centered view that facilitates user/stakeholder discussion focused on further system requirements development and specification.

Background

The HLCD process, especially the concept definition, is a useful tool for establishing a common framework or construct early in the systems engineering and product development cycle. (*Note:* Don't exclude elements in the initial concept that may be beyond the scope of the system or product eventually specified. They may help stimulate thinking.) Though seemingly an obvious initial step in the solution development process and basic systems engineering, frequently the clear articulation of a high-level concept definition is omitted because it is believed that such a definition is implicit knowledge among the group (engineers, acquisition professionals, developers, integrators, users, etc.) or because a detailed design solution is "in hand," thus obviating the need for the higher level composition.

Given the diverse experiences of a typical team, the assumption that even a small number of engineering, acquisition, development, and integration professionals share a common understanding of a complex system is likely to yield disappointing results. More often, engineering, acquisition, and user perspectives diverge at some point, and failure to tie solution development to a common view (the conceptual definition) may allow these differing ideas to go unchallenged and lead to significant disagreements and capability, schedule, and cost impacts later in the design cycle.

Proceeding directly to a more detailed design solution and bypassing an analysis of the trade space performed as an important step in developing the HLCD can lead to an expeditious, but inefficient solution. More effort and resources may eventually be expended to adapt the proposed solution to the user needs, due to discoveries late in the systems engineering process.

In either case, the result can be a solution requiring extensive rework to meet the basic user expectations—often at considerable cost and delivery delay.

Conceptual Definition Process

As part of the early life-cycle systems engineering process, HLCD uses the operational needs assessment, concept of operations (CONOPS), operational requirements, initial capability statements, articulated high-level stakeholder requirements, and an understanding of the domain to lay the foundation for a definition of user expectations and a further understanding of the solution space. The process of HLCD involves a set of steps for translating capability statements or operational requirements into a recognizable concept or model. The process begins by identifying the central capabilities or main objectives of the effort, and proceeds by organizing a set of descriptors aimed at helping illustrate critical attributes of the central objectives. Throughout this process, a deeper understanding of the users and their requirements is developed and captured in the outline or model that characterizes the concept definition; this will later support further system design.

The form this outline or model captures can vary, and depends on many factors, including the complexity of the concept and the breadth of the stakeholder community. One form of this outline or model is a conceptual definition map, shown in Figure 1. This map helps the SE explore a spectrum of factors that must be considered to fully chart user expectations and translate them into a concise definition.

Step-by-step considerations for completing this map (see Figure 1) include:

1. Begin by capturing the main objective(s) or necessary capabilities (green box). These concise statements explain the need, which is clearly defined from the user point of view. They are written for the broader stakeholder and acquisition (including systems engineering) communities.
2. Proceed to identify stakeholders and their roles and objectives (blue box). The MITRE SE will need to know or ascertain who will be involved in the various aspects of the whole solution across development, use, modification, and sustainment of the systems and capabilities supporting the objectives in Step 1. This step also identifies the stakeholder community that will use the concept definition as the basis for exploration of the solution space, and eventual decisions on program direction, acquisition, and further solution development.
3. Describe the key properties of the concept (orange box). These meaningful statements describe the basic properties of the concept so that the full stakeholder community can easily and uniformly understand the needs and objectives of the users.
4. Identify the products, information, or consumables required to meet user requirements (aqua box). These items should tie into the needs of the user and stakeholder community and support the CONOPS and concept of employment.
5. Describe major technical, operational, and organizational interfaces (e.g., to other products, systems, domains, data/information, or communities) (yellow box). This portion of the map describes how the concept and user fit into the large enterprise or interact with other elements of the domain or mission area.

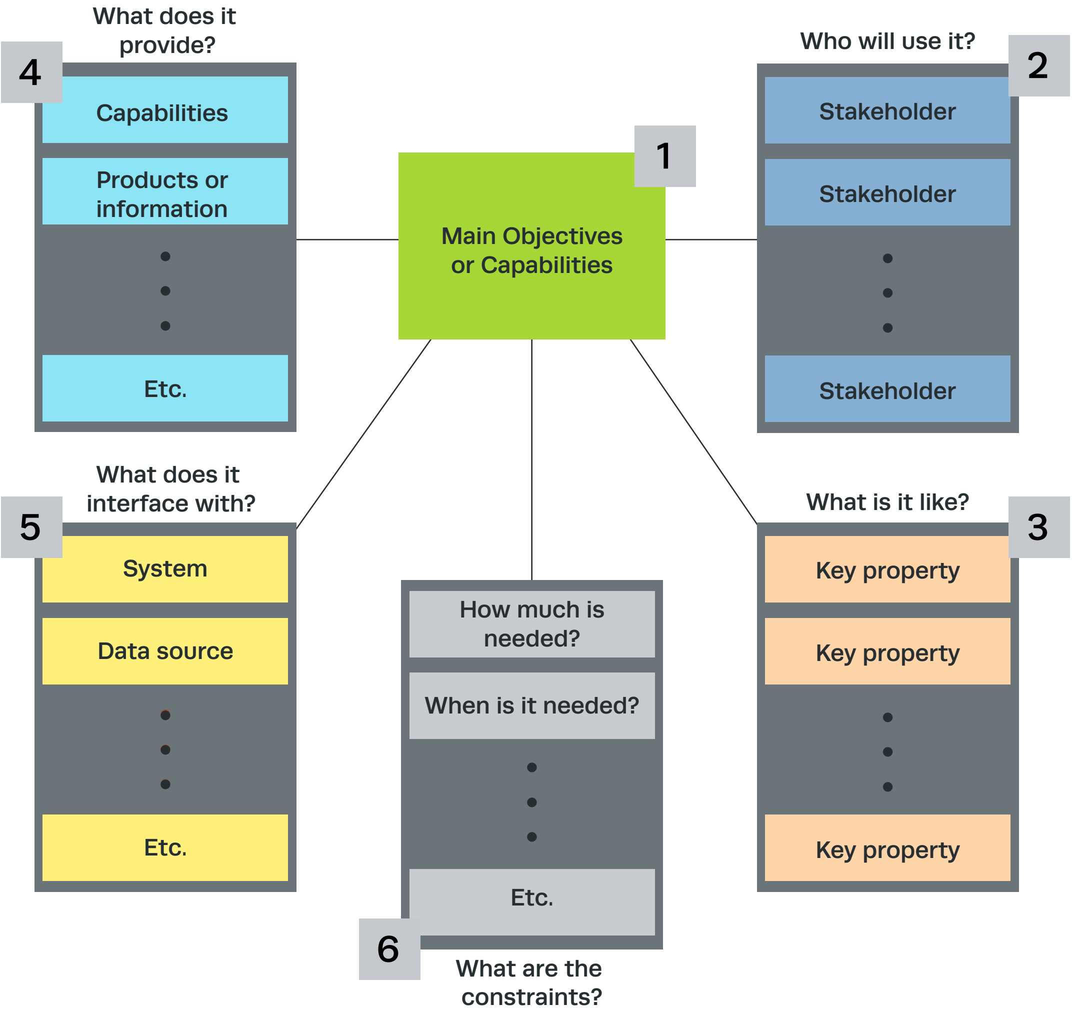


Figure 1. Conceptual Definition Map

6. Articulate constraints (gray box). Describe all constraints, especially those that may help bound the solution space during later design evolution. In particular, cost or schedule constraints may be driving factors in defining the extent of possible future solutions. It is not the intent during this portion of the systems engineering process to eliminate specific solutions, but to express those constraints that may be critical in shaping the future solution.

Best Practices and Lessons Learned

As with any element of systems engineering, potential hazards must be negotiated when applied to complex integrated systems. These include:

Narrowing the solution space too quickly. Becoming too focused on a single approach, technology, or solution during concept definition and eliminating viable (and possibly better) alternatives early in the process.

Narrowing the solution space too slowly. Too much exploration of alternative approaches, excessive waiting for technologies to mature, or working to an expectation of finding a solution that addresses all the consideration of all the stakeholders can lead to analysis paralysis and failure to deliver in a reasonable time period.

Insufficient stakeholder engagement. Failing to fully engage the end-user/stakeholder community, and missing critical perspectives and inputs that might shape the final concept definition. In particular, be sure to engage those who are not immediate stakeholders (e.g., certification and accreditation authorities) whose considerations can be showstoppers if they are engaged late in the process.

Excluding non-materiel solutions. Beware of the inclination to narrow the focus of the high-level concept to materiel options, intentionally or unintentionally avoiding other elements of the possible solution including doctrine, training, operations, etc.

Finally, each concept definition phase provides a new opportunity to ensure a clear, understandable representation of the users' objectives and needs that are developed and vetted by the stakeholder community. A successful concept definition activity helps anchor the future design and engineering efforts, so that customer expectations and acquisition commitments are well managed from the beginning.

References and Resources

Kossiakoff, A., and W. Sweet, 2003, *Systems Engineering: Principles and Practice,* Hoboken, NJ: John Wiley & Sons, Inc.

Murch, R., 2001, *Project Management: Best Practices for IT Professionals,* Upper Saddle River, NJ: Prentice Hall PTR.

Requirements Engineering

Definition: *A requirement is a singular documented need—what a particular product or service should be or how it should perform. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user. Requirements engineering is the discipline concerned with establishing and managing requirements. It consists of requirements elicitation, analysis, specification, verification, and management.*

Keywords: *analysis, definition, development, elicitation, management, requirements, systems engineering, verification*

Context

Requirements are derived from operational needs and concepts and are used as inputs to design and development. Requirements are also an important input to verification, since tests must trace back to specific requirements to determine if the system performs as intended. Requirements indicate what elements and functions are necessary for the particular project. The typical phases of requirements development are eliciting, collecting and developing, analyzing and defining, and communicating and managing requirements. Because of the rapid changes in operational requirements and the pace of technology, increasingly SEs are faced with unprecedented levels of uncertainty in developing requirements.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to integrate business, mission, and operational needs and transform these needs into system requirements. They elicit, develop, analyze, communicate, and manage requirements as well as facilitate stakeholder engagement and agreement on system requirements. They are expected to be able to decompose operational needs and requirements and flow them down to operational capabilities, technical requirements, technical implementation, and verification of the requirements. MITRE SEs are expected to ensure operational value and traceability from the operational need all the way to the system verification and ultimately to the fielding and sustainment of the system. They are expected to actively mitigate uncertainty in requirements through prototyping and experimentation activities.

Discussion

The articles in this topic address the major phases of requirements engineering.

Requirements engineering starts early in concept development by eliciting and collecting operational needs from the relevant user community, and developing requirements from the needs. It involves more than talking to the user or reading their concept of operations, and asking them to review the requirements you created. It is a disciplined approach that includes collecting, validating, prioritizing, and documenting requirements. The article “Eliciting, Collecting, and Developing Requirements” describes a disciplined approach that can be used for different types of strategies, from classic large-scale Department of Defense block acquisitions to agile incremental acquisitions.

Toward the end of the eliciting and collecting phase, SEs analyze the requirements to ensure they are sound and can form a stable basis for the duration of the planned development and testing period. The article “Analyzing and Defining Requirements” describes attributes of a well-crafted requirement to minimize design missteps, confusion, and re-work downstream. It also references tools that exist within MITRE to support and manage this phase of the requirements engineering effort.

Despite best efforts, sometimes the requirements management techniques described in the previously mentioned articles are insufficient. This occurs most often when the user is unsure of their needs or leading-edge technology is needed to meet requirements. In this environment, key tools in the MITRE SE’s toolbox are prototyping and experimentation. These are particularly useful for gauging whether a requirement is achievable or assessing feasibility and maturity of a technology to meet a requirement. The article “Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation” discusses when and how these tools should be applied, the different approaches, “weight” or fidelity available (and which level makes sense for what situations), and ideas for how to evolve the prototyping and experimentation efforts over time to reduce the risk of requirements uncertainty.

References and Resources

International Council on Systems Engineering (INCOSE) website, http://www.incose.org/.

Kossiakoff, A., W. N. Sweet, S. Seymour, and S. M. Biemer, 2011, *Systems Engineering: Principles and Practice,* 2nd Ed., Wiley.

Stevens, R., P. Brook, K. Jackson, and S. Arnold, 1998, *Systems Engineering: Coping with Complexity,* Prentice Hall.

Sutcliffe, A., 1996, "A Conceptual Framework for Requirements Engineering," *Requirements Engineering Journal*, Vol. 1, No. 3, Springer, London.

"Systems Engineering," Wikipedia, accessed March 11, 2010.

The MITRE Corporation, "Requirements Engineering," Systems Engineering Competency Model, section 2.2, accessed February 4, 2010.

U.S. Department of Homeland Security, Requirements Engineering, accessed March 11, 2010.

Definition: *Requirements define the capabilities that a system must have (functional) or properties of that system (non-functional) that meet the users' needs to perform a specific set of tasks (within a defined scope).*

Keywords: *agile, elicitation, elicitation techniques, project scope, requirements, requirements attributes, requirements elicitation, root cause, scope, spiral, stakeholders, user requirements, users, waterfall*

REQUIREMENTS ENGINEERING

Eliciting, Collecting, and Developing Requirements

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to elicit business, mission, and operational needs from operational users and other stakeholders. They are also expected to be able to analyze, integrate, and transform these needs into system requirements as well as facilitate stakeholder engagement on and resolution of requirements. MITRE SEs are expected to be able to tailor the principles of requirements elicitation to different development methodologies (waterfall, spiral, agile, etc.).

Overview

After operational needs are assessed and the concept of operations (CONOPS) and high-level concept definition are completed, the next step—and typically the first task on development projects—is to discover, elicit, collect, define, and analyze requirements. Requirements will cover various aspects of a capability or system—user needs, behavioral, quality, implementation, etc. Given these, the SE will analyze, transform, and integrate users' needs into system requirements. For more information on the first steps in development projects, see the SEG's Concept Development topic.

Figure 1 highlights a typical process for collecting and evaluating requirements. Allocating sufficient time and effort to the requirements process to build a strong foundation for the effort has proven to be cost-effective in the long run.

Figure 1 represents typical sequencing of many of the activities and milestones that are part of the requirements collection and management processes. Activities may be added, modified, deleted, and their sequences changed, depending on the scope and type of project

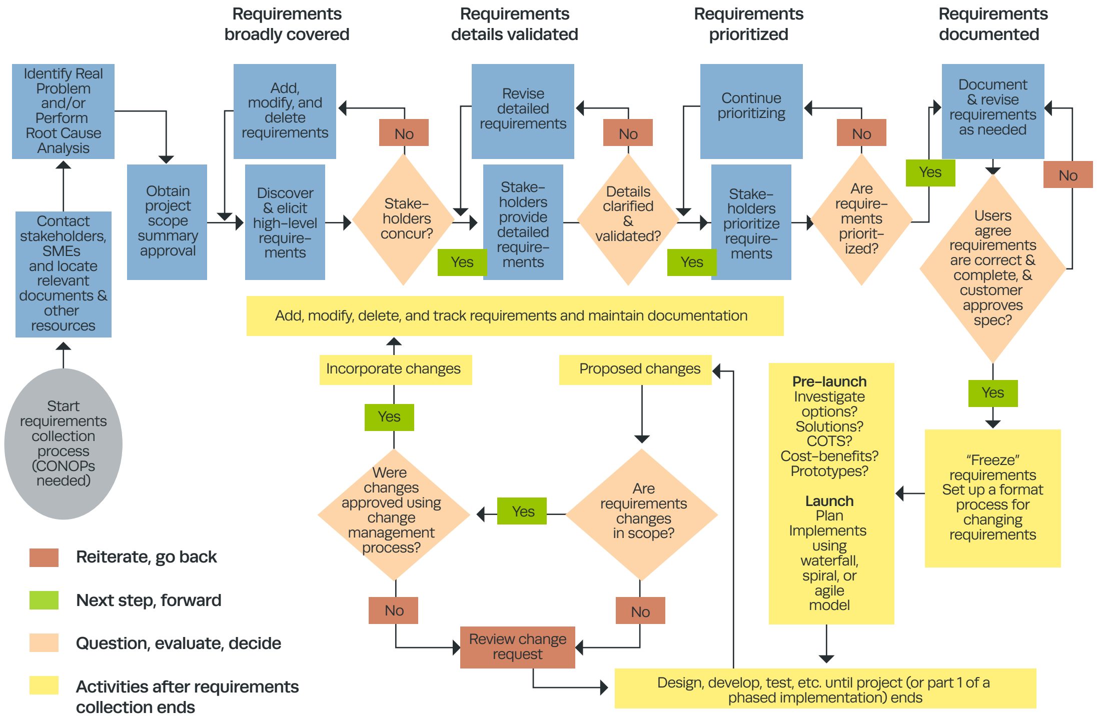

Requirements broadly covered
Requirements details validated
Requirements prioritized
Requirements documented
Identify Real Problem and/or Perform Root Cause Analysis
Contact stakeholders, SMEs and locate relevant documents & other resources
Start requirements collection process (CONOPs needed)
Obtain project scope summary approval
Add, modify, and delete requirements
Discover & elicit high-level require-ments
No
Stake-holders concur?
Yes
Revise detailed requirements
Stake-holders provide detailed require-ments
No
Details clarified & validated?
Yes
Continue prioritizing
Stake-holders prioritize require-ments
No
Are require-ments priorit-ized?
Yes
Document & revise requirements as needed
No
Users agree requirements are correct & complete, & customer approves spec?
Yes
"Freeze" requirements Set up a format process for changing requirements
Add, modify, delete, and track requirements and maintain documentation
Incorporate changes
Yes
Were changes approved using change management process?
No
Proposed changes
Yes
Are requirements changes in scope?
No
Review change request
Pre-launch
Investigate options? Solutions? COTS? Cost-benefits? Prototypes?
Launch
Plan Implements using waterfall, spiral, or agile model
Design, develop, test, etc. until project (or part 1 of a phased implementation) ends
🟥 Reiterate, go back
🟩 Next step, forward
🟧 Question, evaluate, decide
🟨 Activities after requirements collection ends

Figure 1. Overview of Requirements Collection and Change Processes

or task. Generally, subtasks within a larger project focus on fewer activities and may have different stakeholders and finer grained criteria for success than the project itself.

The process the SE follows depends on the project's complexity and implementation methodology: waterfall, spiral, agile, etc. Studies have shown that accurate, well-defined, and clearly stated requirements reduce development time and effort and are essential to the quality and success of the final product. Users provide functional and nonfunctional requirements, which form the substrate on which the project is built. Functional requirements are associated with the capability/application need to directly support the users' accomplishment of their mission/tasks (features, components, etc.). Performance requirements are those that are typically implicit and technical in nature that emerge as system requirements to satisfy the users' functional needs (e.g., quality of service, availability, timeliness, accuracy). SEs work closely with users to observe, discuss, and understand the user requirements.

- **Waterfall model:** Projects using the waterfall model progress through a series of phases/milestones in a linear fashion, with the first phase dedicated to the requirements task. The first milestone occurs when a complete set of functional, performance, and other requirements has been documented, validated, and approved by the user. Stabilizing requirements early in the project's life cycle facilitates subsequent project work and significantly reduces risk. This type of model can be feasible in the increasingly rare situations when the customer mission or business is fairly static, the need is focused, and the user environment is stable.
- **Spiral model:** Each cycle or level in the spiral model includes several activities found in various phases of the waterfall model. This model is used to reduce project risk incrementally. At the end of each cycle, stakeholders analyze risks and develop appropriate risk reduction strategies for use at the next level. The SE collects, documents, and updates requirements before the project starts and after each cycle. The requirements may be known up front, but spirals are used to get capabilities to the users quicker, or the requirements may not be completely known up front, but the basic operational needs and concepts are known, so projects can begin and allow the future evolution's requirements to be determined over time. The first milestone usually occurs early in the spiral under these conditions: requirements for the spiral are complete and agreed to by the user concurrently with an operational concept, plans, design, and code.
- **Agile model:** The agile software development model does not require detailed documentation and design at start-up but does require flexible systems engineering support during the project. Typically small efforts are performed and the set of requirements is focused on small, specific capabilities with the users and developers teaming to work the interplay of requirements and capabilities together. "Agile" emphasizes very short cycles, substantial user collaboration from start to finish, close teamwork, constant

communication among participants, the ability to adapt to change, and incremental development. The goal is to quickly develop working functional software that meets the users' needs, not produce detailed requirements or documentation. The SE may wear several hats in an agile environment by providing support as needed, for example: identifying emerging requirements that may violate standards and regulations; analyzing, then documenting requirements as they evolve; calculating metrics; and writing functional specifications, test cases, meeting minutes, and progress reports.

- **Using multiple models:** More than one model can be used during a project's development. Regardless of the particular model, all approaches should include requirements elicitation in some form. The activities in the Best Practices (below) are often associated with the waterfall model, but many are modified for use with other models as well. SEs may change, reorder, repeat, or omit activities on the list, depending on the project type, complexity, methodology, and environment. A structured approach can help guide the requirements collection process from the first (i.e., "kickoff") meeting between the SE and stakeholders until requirements are baselined and approved. These guidelines are applicable and adaptable for requirements collection on large and small systems, new systems, and existing systems that are being updated or replaced. Requirements may also evolve over time due to mission changes, business environment changes, etc. The requirements must be managed throughout the life cycle to ensure the needed capabilities are being created and delivered to accommodate changes.

Challenges exist today with the requirements engineering process—frequently, sufficient time is not allocated to understand operational concepts and thus the requirements associated with them; requirements are specified, not managed to accommodate changes; requirements are not revisited often enough to further assess trade-offs that users would consider in order to manage schedule and costs. However, a good requirements process can provide a strong foundation for satisfying user needs.

Best Practices

Apply good interpersonal skills. Such skills are always an asset, but they are a necessity when eliciting requirements. When SEs are objective and open-minded and have good listening skills, their relationships with users and other team members are productive. Their ability to effectively communicate project status and resolve issues and conflicts among stakeholders increases the likelihood of the project's success.

Think broadly. SEs with broad knowledge of the enterprise in which requirements are being developed (whether for a system, service, or the enterprise) add value and may be able to identify solutions (e.g., process changes) that are cost-effective.

Be prepared. Collect data and documents that provide context for the project. Review data generated during enterprise and concept analysis, and review any business case and decision briefings for the project. Become familiar with historical information, organizational policies, standards, and regulations that may affect requirements and impose constraints. Gather information on previous projects, successful or not, that share characteristics with the new project. Review their system specifications and other technical documents, if they exist. The SE may derive "explicit" or "implicit" lessons learned and requirements from data on the previous project. Find out whether there are descriptions of current operations, preferably an approved concept of operations (see the SEG's Concept Development topic), and any documented issues. Some of this material may identify potential stakeholder types and subject matter experts (SMEs) that may be needed. Draft a requirements collection plan, estimate resources needed, and consider the types of tools that would be appropriate on a project using this particular methodology. Identify potential risks that might arise during the requirements collection process (e.g., key stakeholders are unavailable due to time constraints) and plan risk mitigation strategies.

Identify and manage stakeholders. A single individual or organization often initiates a project. Inevitably, the new project will affect other individuals, organizations, and systems, either directly or indirectly, thereby expanding the list of stakeholders. Stakeholders' "roles" are: the executive sponsor funding the project and possibly a contributor to requirements; primary stakeholders and others providing functional and performance requirements; stakeholders affected by the project indirectly (e.g., interfacing businesses and operations) who may contribute requirements; SMEs (e.g., managers, system architects and designers, security staff, and technical and financial experts); and stakeholders who must be kept in the loop (e.g., business analysts, legal and financial experts). As needed, stakeholders should be asked to review, comment on, and approve requirements for which they are responsible. Set up a process for communicating with stakeholders (e.g., meetings of all types, formal presentations, biweekly reports, and email).

Determine the root cause of the problem. Before requirements collection starts, it is critical that the SE answer the question: what is the real need that the project and its product are intended to address? The SE must tread carefully but resolutely in the user's environment to uncover the real vs. perceived needs. Examining some of the concept and operational needs information can help with the analysis. The next vital question is: have all stakeholders agreed on a clear and unambiguous need statement that is consistent with the business case? The SE's ability to state the problem in an implementation-independent manner is extremely important. Customers may find it difficult to accept the fact that their solution to their perceived problem is not viable, or that other options should be explored.

Define capability scope. The SE generates a capability scope that provides a framework for the project and guides the requirements collection process. The capability scope is usually the first source of information about the project available to all stakeholders before the project gets under way. It is reviewed by stakeholders and approved by the customers. The SE's goal is to elicit and discover all requirements and ensure that each is within the boundaries described in the scope. This criterion is used to evaluate requirements' changes throughout the life cycle. Scope assessments are not limited to the requirements phase. They are often used to cover project activities from launch to completion, specific activities (e.g., pilots and testing), and for small tasks within larger projects. Capability scopes are generated as needed, for example: before or after requirements collection, or for inclusion in a request for proposal, work breakdown structure, or statement of work. Other documents, such as PDDs (project definition documents) and SOOs (statements of objectives) often serve the same purpose as capability scopes.

Capability scope documents describe the "who, what, when, and why" of the project and include information needed for project planning. Capability scope documents cover most of the topics below. Some top-level information for the scope can be found in the operational needs and concepts information from the user community.

- **Purpose:** What problem is the customer trying to solve? What does the customer need and want? What will this project achieve?
- **Value Proposition:** Why is this capability justified?
- **Objectives/Goals:** High-level goals that can be measured
- **Sponsor:** Who is paying for the capability?
- **Customers:** Who will use the results of the project?
- **Scope of Project:** Activities and deliverables included in this project
- **Out-of-Scope:** Activities and deliverables not included in this project
- **Interfacing:** What are the interfacing capabilities, systems, or user communities that will touch this project?
- **Major Milestones:** Events denoting progress in the project life cycle (e.g., completion of key deliverables or important activities)
- **Dates:** When are deliverables due? What are the planned milestone dates?
- **Critical Assumptions:** Assumptions underlying plans for conducting and completing the project
- **Risks:** Potential changes in the project's environment or external events that may adversely affect the project
- **Issues:** Issues that have already been identified for this project
- **Constraints:** Rules and limitations (e.g., time, resource, funding) that may dictate how the project is carried out
- **Success Criteria:** Outcomes that meet requirements, targets, and goals and satisfy the customer.

Discover and elicit requirements from all relevant sources. The SE collects requirements from many sources including, but not limited to: experienced and new users, other stakeholders, SMEs, managers, and, if necessary, the users' customers. Operational users are key contributors because they provide some or all requirements for the system's functional and performance capabilities and user interface. Their inputs are essential to delivering a product, system, or service that helps improve their efficiency by enabling them to easily access the data they need when they need it. The SE elicits requirements directly or indirectly based on users' informal narratives, observing the user environment, or capturing their responses to targeted questions. The SE wants to learn about the operational users' environments and needs a lot of information for that purpose, such as: detailed descriptions of users' daily, weekly, monthly, and other periodic tasks; documentation (e.g., training manuals); reporting requirements and examples of written reports; preconditions and/or triggers for taking various actions; workflows and sequencing of specific tasks they perform; external and internal rules they must follow including security requirements; interactions with other operational users, staff members, systems, and customers; type and frequency of problems they encounter; and, overall, what does and does not work for them currently. Users' responses to questions such as "describe your ideal system" may open up areas not previously considered. The SE can confirm requirements collected and possibly uncover new ones if given the opportunity to directly observe users doing their jobs. Passive observation is often time well spent.

The SE consults with SMEs to ensure that system, security, and operational requirements are complete and feasible; the SE also brings attention to and incorporates into the requirements, government and other regulations that must be taken into account during the program. Project size and type, complexity, schedule, number of interviewees, and locations are factors that will determine techniques best suited to eliciting requirements for this project. Techniques include direct observation, one-on-one and/or group interviews, brainstorming sessions, focus groups, surveys and targeted questions, and prototyping. Joint (users, developers, integrators, systems engineering) requirements gathering sessions are frequently one of the most powerful techniques for eliciting requirements. When SEs analyze related documents, system interfaces, and data, they are likely to discover new requirements. Reverse engineering may be needed to uncover requirements for legacy systems that are poorly or not documented. Collection activities proceed to the next life-cycle step (e.g., beginning system design) when users confirm that implementation of the current set of requirements will meet their needs, and project staff agrees that they can build a viable product based on these requirements. However, with many changes in the stakeholders' operations, a continuous requirements collection and refinement effort is needed to ensure initial requirements are captured and future capability assessments are started by examining the next evolution of requirements due to change, increased certainty in need, or a phased implementation approach.

Document requirements' types and attributes. Categorizing and organizing many requirements can be daunting. As the process matures, requirements' attributes must be documented and kept up to date to remain traceable during testing, validation, and verification. This process helps SEs and others identify duplicate, missing, and contradictory requirements. Attributes most often tracked are these requirements: ID (number), description, type (e.g., functional, non-functional, performance, explicit, derived, system, operational), priority (e.g., mandatory, desirable), phase (threshold or objective), level of risk, business value (e.g., high, medium, or low), source (e.g., stakeholder, regulation, interface specification), rationale for including the requirement (e.g., improves performance), name of implementer, level of effort, status (e.g., proposed, approved, verified, closed), and, later, release number/release date.

Model requirements for validation. Stakeholders are frequently asked to review documents that include those requirements for which they are responsible. Stakeholders sometimes need help interpreting requirements; they also expect and are entitled to receive clear explanations of outcomes when they are implemented. Explanations can be facilitated by creating "as is" and "to be" process flows, activity diagrams, use cases, entity relationship diagrams, workflow models, and flowcharts. Models can also be in the form of prototypes or experiments to provide a limited functioning context where users can try out various alternatives, and together the user and SE can assess the requirements (see the article "Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation"). Visual aids that are focused on these areas tend to engage the stakeholders' interest. Models show stakeholders how the requirements they contributed represent their statements and goals, and are complete and consistent. Agreeing on the meaning of each requirement and its effect on the final product may call for several iterations of discussions, modifications, and reviews by different groups of stakeholders. Putting everyone on the same page takes time. The SE updates requirements when changes are made at reviews and meetings and tracks issues (e.g., action items) and conflicts. When conflicts cannot be resolved, the SE brings them to the customer's or sponsor's attention, using other levels of MITRE management, as appropriate.

Prioritize requirements. As the collection process winds down, stakeholders are asked to assign a priority to each requirement. There may be differences of opinion about which ones are mandatory, critical, desirable, or optional. It is up to the SE to define each priority (e.g., needs vs. wants), point out inappropriate priorities, and suggest changes based on knowledge of this particular project and past experience. Prioritization ends when stakeholders reach agreement. Getting stakeholders to reach agreement can be difficult. A best practice is to develop and put in place a stakeholder contention adjudication protocol early in the requirements elicitation process. Identifying critical requirements is particularly important when evaluating competing systems and commercial-off-the-shelf products. They are also used to evaluate deliverables at various milestones or evolutions, and, in projects developed incrementally, help determine which requirements are included in each phase.

Work toward getting final agreement from contributing stakeholders. At the end of the requirements collection process, plan to hold a face-to-face requirements review meeting attended by stakeholders who contributed requirements.

Include project team members if possible. Often, last-minute requirements changes are needed to reach consensus that they are complete and correct. At that point, requirements are considered "baselined," "locked," or "frozen." But be careful—flexibility is needed throughout the life cycle to ensure that system development and implementation does not try to meet the exhaustive set of requirements when earlier delivery or perhaps reduced costs could be achieved (e.g., why the prioritization step is very important).

Document requirements for final approval. The requirements document or specification will dictate much of the project's future work. Before the specification is approved, ask reviewers who know the characteristics of "good" requirements to review it. Good requirements are unique and uniquely identified, necessary, consistent, complete, traceable, testable, implementation-free, attainable, unambiguous, and verifiable. It may be necessary to modify or delete "bad" requirements and their dependents. Dependent requirements are associated requirements, many times implicit or emerging from a functional requirement (e.g., need for data would drive a technical need for some form of database/repository). Final approval is made by the executive sponsors or one of the customers. Inevitably, some requirements in the specification will be misinterpreted by implementers, many of whom may be seeing them for the first time. To avoid or minimize misinterpretations, the SE, optimally together with the user community, must be given time to go over the approved specification with designers, software engineers, quality assurance staff, testers, and others to answer questions. As soon as possible, a close community of users, SEs, designers, developers, integrators, and testers should be formed and maintained.

Capture lessons learned. Although MITRE internal projects differ from MITRE direct customer projects in many ways, there are commonalities when it comes to requirements and lessons learned. When a project ends, project members, including managers, are encouraged to document their experiences, good and bad, using the LAMP (Lessons About My Project) template. A facilitator conducts a "Lessons Learned Review" with project participants and uses the LAMP to identify best practices and processes that need improvement. The LAMP site provides many practical case examples of eliciting requirements.

Even when the requirements are good, complete, and correct, as a project is launched, changes are inevitable. Experience has shown that adding, modifying, and deleting requirements after a project is underway greatly increases cost. A formal requirements management process will help control cost, avoid requirements creep, and ensure end-to-end traceability. However, changes do occur and flexibility is needed to manage the changes while concentrating on delivery of capabilities to users as soon as feasible.

Related articles include "Analyzing and Defining Requirements" and "Stakeholder Assessment and Management."

References and Resources

Ambler, S. W., September 1, 2005, "Seeking Stakeholders," *Doctor Dobbs Digest.*

Ambler, S. W., October 1, 2005, "Requirements Wisdom," *Doctor Dobbs Digest.*

Ambler, S. W., September 16, 2008, "Strategies for Addressing Non-Functional Requirements," *Doctor Dobbs Digest.*

Association for Computing Machinery, ACM On-line Course on Eliciting Requirements, accessed March 3, 2010.

Bahill, T., "Eliciting Requirements in Use Cases and What is Systems Engineering? A Consensus of Senior Systems Engineers," accessed March 3, 2010.

Bruegge, B., A. H. Dutoit, and A. A. Dutoit, October 1999, "*Object-Oriented Software Engineering: Conquering Complex and Changing Systems,*" Pearson Education.

Florence, A., April 2002, "Reducing Risks Through Proper Specification of Software Requirements," *CrossTalk: The Journal of Defense Software Engineering.*

Florence, A., October 2003, "Requirements Validation and Verification," *QAI Journal*.

Florence, A., and W. Bail, 2005, "Effective Requirements Practices," The MITRE Corporation.

Gottesdiener, E., March 2008, "Good Practices for Developing User Requirements©," *CrossTalk: The Journal of Defense Software Engineering.*

The MITRE Corporation, "ISIS Systems Technology and Engineering Projects (iSTEP)," accessed March 3, 2010.

The MITRE Corporation, "Lessons About My Project (LAMP)," accessed March 2, 2010.

Nuseibeh, B., and S. Easterbrook, 2000, *Requirements Engineering: A Roadmap*, Department of Computing, Imperial College of Science, Technology & Medicine, London, UK.

Robertson, J., and S. Robertson, February 2006, Volere Requirements Specification Template.

West Pole, Inc., 1996–2005, Use Case and Interviewing Techniques for Focused Requirements Capture.

Definition: *The engineering analysis that ties the needs of users and other stakeholders to the system to be built in a quantifiable and traceable manner.*

Keywords: *analyze, develop, development methods, measures of effectiveness, measures of performance, performance engineering, requirements*

REQUIREMENTS ENGINEERING

Analyzing and Defining Requirements

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to analyze systems requirements to determine if they can be tested, verified, and/or validated, and are unique, complete, unambiguous, consistent, and obtainable, and to trace all requirements to original business and mission needs. They are expected to review requirements to determine conformance with government policy for developing the system and identify potential integration and interoperability challenges.

Background

How can we judge if a system meets the needs of a user community? One way is to look at the system requirements and compare those statements to how the system was built, installed, and operated. For large enterprise systems, traditionally there has been a substantial lag between requirements definition and field operation of a system. This often affects both the effectiveness of a system and how the system is perceived (e.g., the system is stale and outdated: it does not meet my needs). In part, this lag is addressed by spiral or incremental approaches to enterprise capability development (see the article "Eliciting, Collecting, and Developing Requirements"). The implication for requirements is that they should be defined in detail in time to support the increment or spiral in which they will be developed. This mitigates the problem of locking down a requirement so early that it contributes to the perception of staleness when it is finally delivered. Additionally, when allocating requirements to an increment or spiral, they should be stable enough over the increment's planned development and testing period that the capability can still be expected to meet the user needs when delivered. Beyond that, requirements analysis during concept development must be efficient, accurate, rational, and traceable.

Characteristics of Good Requirements

MITRE SEs encounter many types of projects and systems—from research and development, to technical consulting work, to acquisition. Whatever the context, a good requirements statement typically has these characteristics [1]:

- **Traceable:** A requirement must be traceable to some source such as a system-level requirement, which in turn needs to be traced back to an operational need and be attributable to an authoritative source, whether a person or document. Each requirement should have a unique identifier allowing the software design, code, and test procedures to be precisely traced back to the requirement.
- **Unambiguous:** Test the wording of the requirement from different stakeholders' perspectives to see if it can be interpreted in multiple ways.
- **Specific and singular:** Needed system attributes (e.g., peak load) are described clearly as atomic, singular thoughts.
- **Measurable:** System functions can be assessed quantitatively or qualitatively.
- **Performance specified:** Statements of real-world performance factors are associated with a requirement.
- **Testable:** All requirements must be testable to demonstrate that the end product satisfies the requirements. To be testable, requirements must be specific, unambiguous, and quantitative whenever possible. Vague, general statements are to be avoided.

- **Consistent:** Requirements must be consistent with each other; no requirement should conflict with any other requirement. Check requirements by examining all requirements in relation to each other for consistency and compatibility.
- **Feasible:** It must be feasible to develop software that will fulfill each software requirement. Requirements that have questionable feasibility should be analyzed during requirements analysis to prove their feasibility. If they cannot be implemented, they should be eliminated.
- **Uniquely identified:** Each need is stated exactly once to avoid confusion or duplicative work. Uniquely identifying each requirement is essential if requirements are to be traceable and able to be tested. Uniqueness also helps in stating requirements in a clear and consistent fashion.
- **Design-free:** Requirements should be specified at the requirements level and not at the design level. Describe the requirement functionally from a requirement point of view, not from a design point of view (i.e., describe the functions that the system must satisfy). A requirement reflects "what" the system shall accomplish, while the design reflects "how" the requirement is implemented.
- **Uses "shall" and related words:** In specifications, using "shall" indicates a binding provision (i.e., one the specification users must implement). To state nonbinding provisions, use "should" or "may." Use "will" to express a declaration of purpose (e.g., "The government will furnish…") or to express future tense.

Each of these characteristics contributes to the integrity and quality of the requirements and the system or capability to be developed. Enforcing these characteristics during the requirements activity helps keep the entire development effort organized and reproducible, and avoids issues later in the life cycle. The goal of a requirements process is to define a system or capability that ties the needs of the users and other stakeholders to the system to be built so that it satisfies the needs within a specified schedule and cost, and possesses the required performance characteristics, including characteristics like information assurance, quality, reliability, internationally enabled, and sustainability. Observing the above requirements characteristics will help to maintain engineering rigor, content, and value of the engineering analysis.

Measures Associated with Requirements Analysis [2]

The typical categories of measures associated with determining if a system complies with the requirements include:

- **Measures of Effectiveness (MOEs):** MOEs are measures of mission success stated under specific environmental and operating conditions, from the users' viewpoint. They relate to the overall operational success criteria (e.g., mission performance, safety, availability, and security).

- **Measures of Performance (MOPs):** MOPs characterize specific physical or functional characteristics of the system's operation, measured under specified conditions. They differ from MOEs in that they are used to determine whether the system meets performance requirements necessary to satisfy the MOE.
- **Key Performance Parameters (KPPs):** KPPs are a stakeholder-defined measure that indicates a minimal and critical system performance and level of acceptance.

Best Practices and Lessons Learned

Baseline and agree. Developing requirements is usually a collaborative activity, involving users, developers, maintainers, integrators, etc., so avoid placing the responsibility of requirements analysis solely on one stakeholder. When all team members acknowledge a set of requirements is done, this is called a baseline (realizing that this will evolve—see "Be flexible" below).

Requirements analysis is an iterative process, so plan accordingly. At each step, the results must be compared for traceability and consistency with users' requirements, and then verified with users, or go back into the process for further analysis, before being used to drive architecture and design.

Pay special attention to interface requirements. Requirements must clearly capture all the interactions with external systems and the external environment so that boundaries are clear. Remember that external interfaces can be influenced by the architecture of a system or subsystems. In some cases, hidden external interfaces will be established because internal subsystem-to-subsystem communications use an external medium (e.g., radios connect subsystems via airwaves) or other assets that are not part of the system (e.g., satellite relay, external network). Examples of tools for recognizing external interfaces include the DoD Architecture Framework (DoDAF) operational and system views referred to as OV-1, OV-2, and SV-1 diagrams.

Be flexible. To balance out rigidness of baselining requirements, a development team should consider what constitutes a "change of requirements" as distinguished from a valid interpretation of requirements. The key is to find a balance between adherence to a baseline and sufficient flexibility (e.g., to encourage innovation, support the changing mission).

Use templates and tools that suit your needs. To get started quickly, make use of resources provided by your sponsor organization or internal MITRE resources.

Use storyboarding, use cases, campaign modeling and simulation tools, and other tools that capture users, their activities, and the information flows. Prototyping and experiments are effective ways to collect the information.

MIL-STD-490A, "Specification Practices," is a military standard for defining specification of military systems. Although it is officially cancelled, it provides descriptions of various types of specifications and their contents.

Data Item Description DI-IPSC-81431A, "System/Subsystem Specification," is a template for a traditional military system or subsystem specification. It describes the format and contents of a traditional specification. It often requires tailoring to a particular application. It is a useful tool to stimulate thought on topics to be included in the requirements. There are specialized database tools that are designed to capture and manage requirements. A good source for information on available tools is the INCOSE website [3] under Requirements Management. The list is free to the public.

References and Resources

1. Florence, A., April 2002, "Reducing Risks Through Proper Specification of Software Requirements," *STSC CrossTalk*, Vol. 15, No. 4.
2. Roedler, G. J., and C. Jones, 2005, *Technical Measurement: A Collaborative Project of PSM, INCOSE, and Industry.*
3. INCOSE website, http://www.incose.org/.

Additional References and Resources

Ho, C-W., et al., 2006, "On Agile Performance Requirements Specification and Testing," IEEE Agile Conference.

Kossiakoff, A., and W. Sweet, December, 2002, *Systems Engineering Principles and Practice*, Wiley-Interscience.

Light, M., April 18, 2005, "Agile Requirements Definition and Management Will Benefit Application Development," Gartner RASCore Research Note G00126310, R1741.

Perron, J., Fall 2007, "User-Centered Requirements Analysis: Who, What, When, Where, Why & How," *SEPO Collaborations*, Vol. 5, Issue 2, The MITRE Corporation.

Definition: *Prototyping and experimentation are two closely related methods that can help systems engineers (SEs) drive requirements uncertainty out of the requirements process.*

Keywords: *CONOPS, experimentation, exploration, prototyping, requirements, uncertainty*

REQUIREMENTS ENGINEERING

Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to identify uncertainty in requirements and actively take steps to manage and mitigate it, including considering uncertainty in associated areas such as operational concepts and others (see the SEG's Concept Development topic). MITRE SEs are expected to understand the range and styles of prototyping and experimentation, and the potential impact of each when applied during requirements engineering. SEs are expected to understand the value in having MITRE execute a prototyping activity as opposed to (or in conjunction with) a contractor. They are also expected to be aware of experimental venues, events, and laboratories that exist to support these activities.

Background

Successfully developing systems or capabilities to meet customers' needs requires the ability to manage uncertainty when defining requirements. For example, how will analytical assessments of performance or functional requirements match the reality of their implementation when the system or capability is fielded? What unintended technical, operational, or performance issues are likely to occur? Will technology essential to meet a requirement perform as expected when using realistic user data, or when injected in operational environments and contexts? Are the user concepts of operations really supportable given new technical capabilities? Prototyping and experimentation are two methods that can help address these issues.

Prototyping

Prototyping is a practice in which an early sample or model of a system, capability, or process is built to answer specific questions about, give insight into, or reduce uncertainty or risk in many diverse areas, including requirements. This includes exploring alternative concepts and technology maturity assessments as well as requirements discovery or refinement. It is a part of the SE's toolkit of techniques for managing requirements uncertainty and complexity and mitigating their effects.

The phase of the systems engineering life cycle and the nature of the problem the prototype is intended to address influence the use and type of prototyping. Prototyping may be identified immediately after a decision to pursue a material solution to meet an operational need. In this situation, prototypes are used to examine alternative concepts as part of the analysis of alternatives to explore the requirements space to determine if other approaches can better meet the requirements. Prototyping to explore and evaluate the feasibility of high-level conceptual designs may be performed early in technology development as part of government activities to assess and increase technology maturity, discover or refine requirements, or develop a preliminary design. A prototype may even be developed into a reference implementation—a well-engineered example of how to implement a capability, often based upon a particular standard or architecture—and provided to a commercial contractor for production as a way of clarifying requirements and an implementation approach.

For more information on prototyping, see the article "Competitive Prototyping."

Experimentation

Experimentation adds a component of scientific inquiry to the above, often supported by realistic mission/domain context. Performing experiments with a realistic context allows the evaluator to assess and evaluate hypotheses about concept of operations (CONOPS), feasibility of technology, integration with other systems, services, and data, and other concepts that support requirements refinement. Experimentation environments, or laboratories, allow acquisition personnel, real-world users, operators, and technologists to collaboratively evaluate concepts and prototypes using combinations of government, open source, and commercial-off-the-shelf products. In these environments, stakeholders can evolve concepts and approaches—in realistic mission contexts—and quickly find out what works and what doesn't, ultimately reducing risks by applying what they've learned to the acquisition process.

It is useful to consider three broad stages of experimentation, which form a pipeline (see Figure 1):

- **Lightweight Exploration:** Driven by operator needs, this stage is distinctive for its quick brainstorming and rapid assembly of capabilities with light investment requirements. It allows for a "first look" insight into new concepts and newly integrated capabilities that can support requirements generation. MITRE's ACME (Agile Capability Mashup Environment) Lab is an example of a lightweight experimentation venue.
- **Low/Medium-Fidelity Experimentation:** This stage involves significant engagement with users, operators, and stakeholders, and is typified by human-in-the-loop simulations and possibly real-world capabilities and data with experimental design and attempts to control independent variables. MITRE's Collaborative Experimentation Environment (CEE) and iLab-based "Warfighter Workshops" are two examples of low/medium-fidelity experimentation venues. These venues allow concept exploration and alternative evaluations that can support requirements clarification.
- **High-Fidelity Experimentation:** These experiments are planned with sponsors to refine existing CONOPS that can support requirements refinement. They often feature highly realistic models and simulations of entities, timing, sensors, and communication networks along with some real-world applications. MITRE's Naval C4ISR Experimentation Lab (NCEL) is an example of a high-fidelity experimentation venue.

Prototype solutions from any of the preceding experimental stages can be used to support the requirements management process. Generally the products from the lightweight end of the venue spectrum support the early stages of requirements management (CONOPS, concept development, etc.), whereas those at the high-fidelity end tend to support refinement of relatively mature

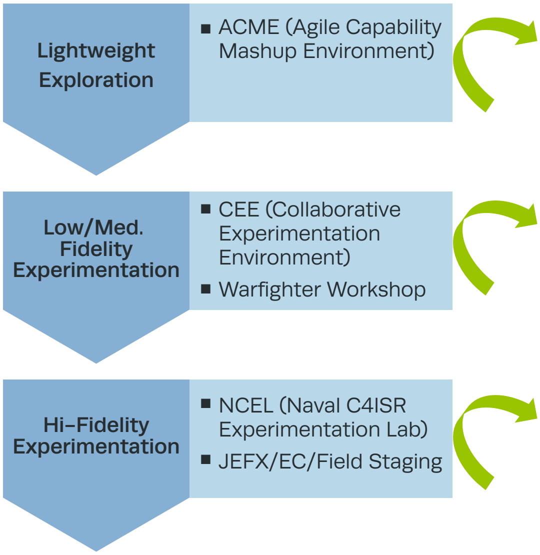

Lightweight Exploration
ACME (Agile Capability Mashup Environment)
Low/Med. Fidelity Experimentation
CEE (Collaborative Experimentation Environment)
Warfighter Workshop
Hi-Fidelity Experimentation
NCEL (Naval C4ISR Experimentation Lab)
JEFX/EC/Field Staging

Figure 1. Experimentation Pipeline

requirements. These solutions can also be transitioned, given appropriate circumstances, to operators in the field, to industry, or to other parties for evaluation or use.

Best Practices and Lessons Learned

Be opportunistic. The acquisition process is structured, linear, and can, in some circumstances, seem to stifle innovation. Embrace requirements uncertainty as an opportunity to inject innovation and think freely.

Act early in the acquisition life cycle. Prototyping early in the acquisition life cycle serves as a method to reduce requirements risk and may be of interest to program managers attempting to avoid late-acquisition-life-cycle change (e.g., requirements creep), especially if there is requirements uncertainty.

Seek early/frequent collaboration among the three critical stakeholders. Purely technical prototyping risks operational irrelevance. It is vital to involve technologists, operators, and acquirers/integrators early and often in prototyping and experimentation dialogs. Operator involvement is particularly critical, especially in solidifying requirements, yet it is often deferred or neglected. Three of the four recommendations from the 2007 Department of Defense Report to Congress on Technology Transition address this collaboration [1]:

> ... *early and frequent collaboration* is required among the developer, acquirer, and user. This early planning can then serve to mitigate the chasm between Technology Readiness Level (TRL) 5 and TRL 7 by identifying technical issues, resource requirements/sources, avoiding unintended consequences, and ultimately gaining the most yield for the science and technology (S&T) investment.
>
> ... if the program manager were to conduct *early and frequent communication* with the developer about user requirements and companion acquisition plans, much of the development risk could be addressed earlier in the process.
>
> ... the pace at which new technologies are discovered/innovated/developed/deployed in the private sector is staggering, and at odds with the linear, deliberate nature of some government acquisitions ... . Finding ways to *include these innovators in our process* could serve both the government and America's economic competitiveness in the world market.

Use realistic data. Prototypes using unrealistic data often result in failure to address the requirements uncertainty or complexity the prototype was designed to examine. MITRE SEs should take every opportunity to capture real data from their work and help build a repository of this data that can be used across MITRE's activities.

Use loose couplers and open standards to isolate requirements changes. Providing loosely coupled data integration points across components in the prototype allows for changes in one area to explore some aspects of the requirements space, while controlling others. Use open standards, including RESTful services, whenever possible. (See articles "Design Patterns," "Composable Capabilities on Demand (CCOD)," and "Open Source Software" in the SEG's Enterprise Engineering section.)

Develop scalable prototypes. Consider the intended operational or systematic use of the prototype being developed. Does the technology scale? Does the operator workload scale? The performance? Do the results collected from the prototype provide the necessary insight to ensure that modifications to requirements are appropriate for the actual full-scale system?

Prefer rapid increments. Execute quick experimental iterations or spirals of prototype capability development with operator involvement to ensure adequate feedback flow and requirements refinement for future spirals.

Look beyond MITRE for resources. MITRE resources on any given program are limited. If the MITRE project resources cannot support a prototyping or experimentation activity by MITRE staff, look to other mechanisms, such as government teaming, the SBIR (Small Business Innovative Research) process, or willing industry participants.

Look beyond MITRE for venues. Consider holding an experiment on-site at a contractor facility, in an operational setting, at a sponsor training facility, or at other locations to reduce real or perceived barriers to participation and to promote awareness of particular stakeholder contexts and points of view.

References and Resources

1. Deputy Under Secretary of Defense, August 2007, DoD Technology Transition Report to Congress.

System Architecture

Definition: *An architecture is "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [1, 2]."*

Keyword: *architecture*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to understand the role that an architecture plays in system development (e.g., conceptualization, development, and certification), the various purposes for architecture, and the different types of architectures. They are also expected to understand various architecture frameworks, models and modeling, views and viewpoints, as well as when and why each would apply. MITRE SEs are expected to understand different architectural approaches and their applications, including the use of architectural patterns.

Context

At this point in the systems engineering life cycle, an operational need has been expressed and turned into a concept and set of operational requirements (see the SEG's Concept Development topic). They are then analyzed and transformed into a set of system requirements (see the SEG's Requirements Engineering topic). The next step is to develop an architecture (or update an existing architecture for fielded systems) as a basis or foundation to guide design and development.

The article "Architectural Frameworks, Models, and Views" discusses the ways in which an architecture can be described. Various frameworks are used in different domains. Two well-known examples are the Zachman framework and the Department of Defense Architecture Framework. Whatever their specific form, all frameworks focus on defining a set of models, views, and viewpoints to support a range of systems engineering and program management activities and decisions across the system life cycle.

The article "Approaches to Architecture Development" provides an overview of ways to tailor and apply architecture approaches, process, and methodologies to support decision making.

Architectural patterns are a method of arranging blocks of functionality. They can be used at the subsystem (component), system, or enterprise level. The article "Architectural Patterns" describes patterns and discusses how they can simplify and expedite the development process.

Architecture Best Practices

Ensure purpose before architecting.

Ensure that stakeholders have an opportunity to vet architectural trade-offs as they occur.

Evaluate the architecture throughout system development. Although an architecture is intended to be a persistent framework during the life cycle (and life) of a system, unforeseen changes (e.g., new missions) can influence the best of "first version" architectures.

Construct the architecture to help understand technology readiness and evolution, and avoid getting locked in to proprietary or potentially obsolete technologies or captured by a specific vendor.

References and Resources

1. ANSI/IEEE 1471-2000, "Recommended Practice for Architecture Description of Software-Intensive Systems."
2. ISO/IEC 42010:2007, "Systems and Software Engineering—Recommended Practice for Architectural Description of Software-Intensive Systems."

Additional References and Resources

"Architecture," MITRE Systems Engineering Competency Model, vol. 1.13, section 2.3, accessed February 23, 2010.

Bass, L., P. Clements, and R. Kazman, December 30, 1997, *Software Architecture in Practice*, 1st Ed., Addison-Wesley Professional.

"System Architecture," MITRE Project Leadership Handbook.

SYSTEM ARCHITECTURE

Architectural Frameworks, Models, and Views

Definition: *An architecture framework is an encapsulation of a minimum set of practices and requirements for artifacts that describe a system's architecture. Models are representations of how objects in a system fit structurally in and behave as part of the system. Views are a partial expression of the system from a particular perspective. A viewpoint is a set of representations (views and models) of an architecture that covers a stakeholder's issues.*

Keywords: *architecture, architecture description, architecture frameworks, models, viewpoint, views*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to assist in or lead efforts to define an architecture, based on a set of requirements captured during the concept development and requirements engineering phases of the systems engineering life cycle. The architecture definition activity usually produces operational, system, and technical views. This architecture becomes the foundation for developers and integrators to create design and implementation architectures and views. To effectively communicate and guide the ensuing system development activities, MITRE SEs should have a sound understanding of architecture frameworks and their use, and the circumstances under which each available

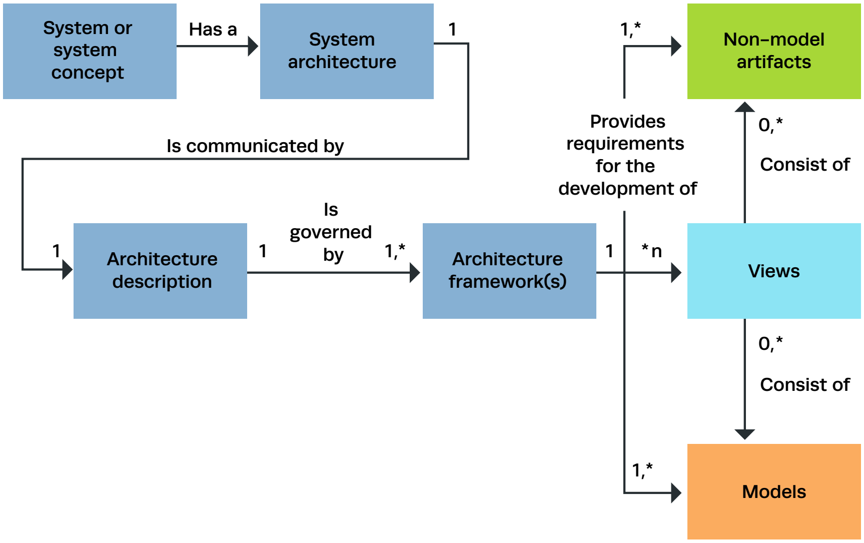


Figure 1. Architecture Framework, Models, and Views Relationship [1]

framework might be used. They also must be able to convey the appropriate framework that applies to the various decisions and phases of the program.

Getting Started

Because systems are inherently multidimensional and have numerous stakeholders with different concerns, their descriptions are as well. Architecture frameworks enable the creation of system views that are directly relevant to stakeholders' concerns. Often, multiple models and non-model artifacts are generated to capture and track the concerns of all stakeholders.

By interacting with intra- and extra-program stakeholders, including users, experimenters, acquirers, developers, integrators, and testers, key architectural aspects that need to be captured and communicated in a program are determined. These architecture needs then should be consolidated and rationalized as a basis for the SE's recommendation to develop and use specific models and views that directly support the program's key decisions and activities. Concurrently an architecture *content and development* governance structure should be developed to manage and satisfy the collective needs. Figure 2 highlights the architecture planning and implementation activities.

MITRE SEs should be actively involved in determining key architecture artifacts and content, and guiding the development of the architecture and its depictions at the appropriate

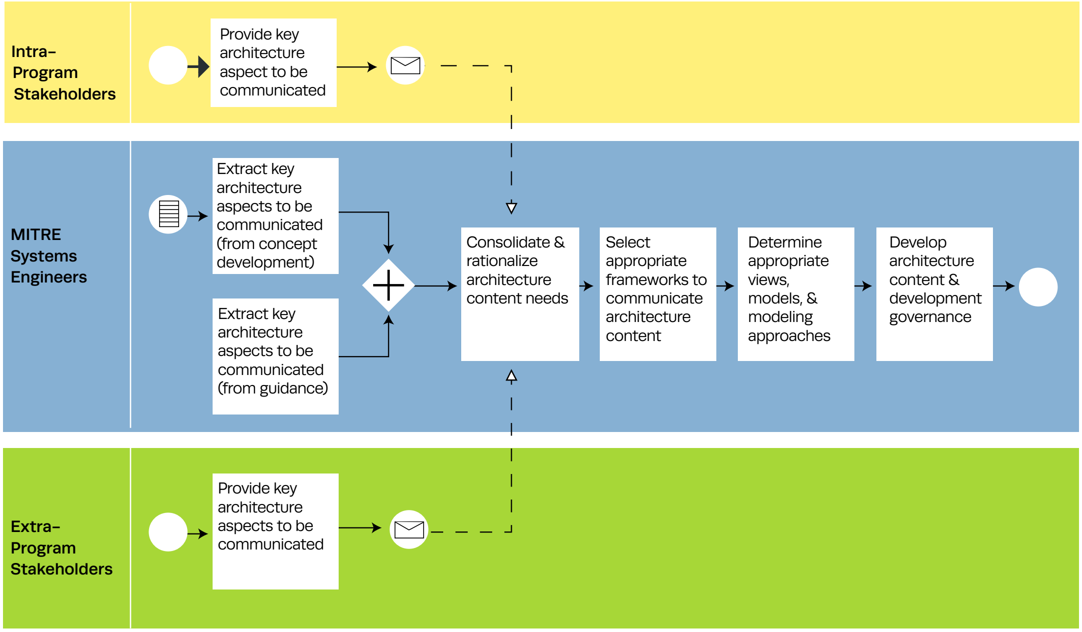

Intra-Program Stakeholders
Provide key architecture aspect to be communicated
MITRE Systems Engineers
Extract key architecture aspects to be communicated (from concept development)
Extract key architecture aspects to be communicated (from guidance)
Consolidate & rationalize architecture content needs
Select appropriate frameworks to communicate architecture content
Determine appropriate views, models, & modeling approaches
Develop architecture content & development governance
Extra-Program Stakeholders
Provide key architecture aspects to be communicated

Figure 2. Architecture Planning and Implementation Activities

levels of abstraction or detail. MITRE SEs should take a lead role in standardizing the architecture modeling approach. They should provide a “reference implementation” of the needed models and views with the goals of: (1) setting the standards for construction and content of the models, and (2) ensuring that the model and view elements clearly trace to the concepts and requirements from which they are derived.

Determining the Right Framework

Though many MITRE SEs have probably heard of the Department of Defense Architecture Framework (DoDAF), other frameworks should be considered. Figure 3 shows that an SE working at an enterprise level should also be versed in the Federal Enterprise Architecture Framework (FEAF). To prevent duplicate efforts in describing a system using multiple frameworks, establish overlapping description requirements and ensure that they are understood among the SEs generating those artifacts. The article “Approaches to Architecture Development” details the frameworks.

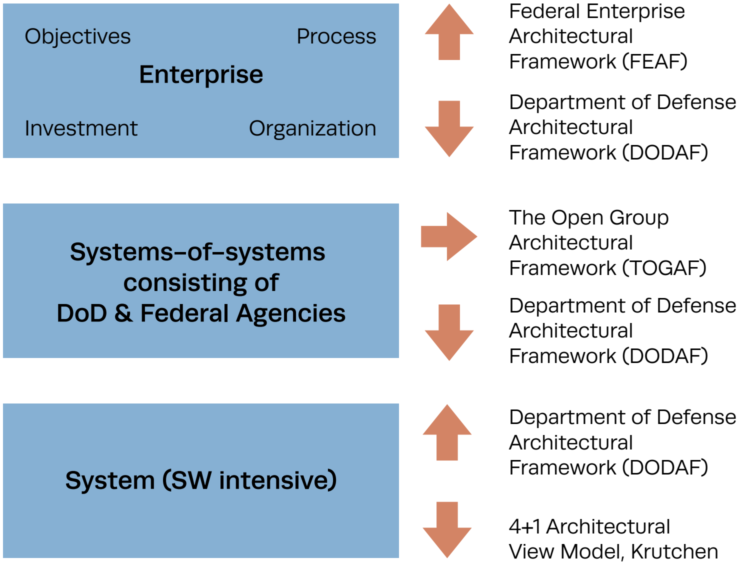

Objectives
Process
Enterprise
Investment
Organization
Federal Enterprise Architectural Framework (FEAF)
Department of Defense Architectural Framework (DODAF)
Systems-of-systems consisting of DoD & Federal Agencies
The Open Group Architectural Framework (TOGAF)
Department of Defense Architectural Framework (DODAF)
System (SW intensive)
Department of Defense Architectural Framework (DODAF)
4+1 Architectural View Model, Krutchen

Figure 3. Applying Frameworks

Best Practices and Lessons Learned

A program may elect not to use architectural models and views, or elect to create only those views dictated by policy or regulation. The resources and time required to create architecture views may be seen as not providing a commensurate return on investment in systems engineering or program execution. Consider these cultural impediments. Guide your actions with the view that architecture is a tool that enables and is integral to systems engineering. Consider the following best practices and lessons learned to make architectures work in your program.

Purpose is paramount. Determine the purpose for the architecting effort, views, and models needed. Plan the architecting steps to generate the views and models to meet the purpose only. Ultimately models and views should help each stakeholder reason about the structure and behavior of the system or part of the system they represent so they can conclude that their objectives will be met. Frameworks help by establishing minimum guidelines for each stakeholder's interest. However, stakeholders can have other concerns, so use the framework requirements as discussion to help uncover as many concerns as possible.

A plan is a point of departure. There should be clear milestone development dates, and the needed resources should be established for the development of the architecture views and models. Some views are precursors for others. Ensure that it is understood which views are "feeds" for others.

Know the relationships. Models and views that relate to each other should be consistent, concordant, and developed with reuse in mind. It is good practice to identify the data or information that each view shares, and manage it centrally to help create the different views. For guidance on patterns and their use/reuse, see the article "SEG Architectural Patterns."

Be the early bird. Inject the idea of architectures early in the process. Continuously influence your project to use models and views throughout execution. The earlier the better.

No one trusts a skinny cook. By using models as an analysis tool yourself, particularly in day-to-day and key discussions, you maintain focus on key architectural issues and demonstrate how architecture artifacts can be used to enable decision making.

Which way is right and how do I get there from here? Architectures can be used to help assess today's alternatives and different evolutionary paths to the future. Views of architecture alternatives can be used to help judge the strengths and weaknesses of different approaches. Views of "as is" and "to be" architectures help stakeholders understand potential migration paths and transitions.

Try before you buy. Architectures (or parts of them) can sometimes be "tried out" during live exercises. This can either confirm an architectural approach for application to real-world situations or be the basis for refinement that better aligns the architecture with operational reality. Architectures also can be used as a basis for identifying prototyping and experimentation activities to reduce technical risk and engagements with operational users to better illuminate their needs and operational concepts.

Taming the complexity beast. If a program or an effort is particularly large, models and views can provide a disciplined way of communicating how you expect the system to behave. Some behavioral models such as business process models, activity models, and sequence diagrams are intuitive, easy to use, and easy to change to capture consensus views of system behavior. For guidance on model characterization, see the article "Approaches to Architecture Development."

Keep it simple. Avoid diagrams that are complicated and non-intuitive, such as node connectivity diagrams with many nodes and edges, especially in the early phases of a program. This can be a deterrent for the uninitiated. Start with the operational concepts, so your architecture efforts flow from information that users and many other stakeholders already understand.

Determining the right models and views. Once the frameworks have been chosen, the models and views will need to be determined. It is not unusual to have to refer to several sets of guidance, each calling for a different set of views and models to be generated.

But it looked so pretty in the window. Lay out the requirements for your architectures—what decisions it supports, what it will help stakeholders reason about, and how it will do so. A simple spreadsheet can be used for this purpose. This should happen early and often throughout the system's life cycle to ensure that the architecture is used. .

How do I create the right views? Selecting the right modeling approach to develop accurate and consistent representations that can be used across program boundaries is a critical systems engineering activity. Some of the questions to answer are:

- Is a disciplined architecture approach embedded in the primary tool my team will be using, as in the case of Activity-Based Modeling (ABM) being embedded in system architecture, or do we have to enforce an approach ourselves?
- Are the rules/standards of the modeling language enforced in the tool, as in the case of BPMN 2.0 being embedded in iGrafix?
- Do I plan to generate executable models? If so, will my descriptions need to adhere to strict development guidelines to easily support the use of executable models to help reason about performance and timing issues of the system?

Bringing dolls to life. If your program is developing models for large systems supporting missions and businesses with time-sensitive needs, insight into system behavior is crucial. Seriously consider using executable models to gain it. Today, many architecture tools support the development of executable models easily and at reasonable cost. Mission-Level Modeling (MLM) and Model Driven or Architecture-Based/Centric Engineering [2] are two modeling approaches that incorporate executable modeling. They are worth investigating to support reasoning about technology impacts to mission performance and internal system behavior, respectively.

How much architecture is enough? The most difficult conundrum when deciding to launch an architecture effort is determining the level of detail needed and when to stop producing/updating artifacts. Architecture models and views must be easily changeable. There is an investment associated with having a "living" architecture that contains current information, and differing levels of abstraction and views to satisfy all stakeholders. Actively discuss this sufficiency issue with stakeholders so that the architecture effort is "right-sized." See the Architecture Specification for CANES [3].

Penny wise, pound foolish. Generating architecture models and views can seem a lot easier to not do. Before jumping on the "architecture is costly and has minimal utility" bandwagon, consider these questions:

- Will there be a need to tell others how the system works?
- Will there be a need to train new personnel on a regular basis (every one to three years) in system operations?
- Will there be a need to tell a different contractor how the system works so that costs for maintaining and refreshing the system remain competitive?

- Will there be a need to assess the system's viability to contribute to future mission needs?

If you answer yes to one or more of these questions, consider concise, accurate, concordant, and consistent models of your system.

References and Resources

1. IEEE Standard 1471, accessed February 26, 2010.
2. Wheeler, T., and M. Brooks, 2006, Experiences in Applying Architecture-Centric Model Based Systems engineering to Large-Scale, Distributed, Real-Time Systems, The MITRE Corporation.
3. Navy PMW 160 Tactical Networks, May 20, 2009, "Architecture Specification for Consolidated Afloat Network and Enterprise Services (CANES), Increment 1."

Additional References and Resources

CIO Council, September 1999, "Federal Enterprise Architecture Framework."

DoDAF Architecture Framework, ver. 2.0, 2008.

Krutchen, P., 1995, "Architectural Blueprints—The '4+1' View Model of Software Architecture."

Ring, S. J., et al., 2004, "An Activity-based Methodology for Development and Analysis of Integrated DoD Architectures."

The Open Group Architecture Framework (TOGAF), version 9.

Definition: *Architecture development can be thought of as both a process and a discipline that aids the development of mission-effective systems.*

Keywords: *DoDAF 6-step architecture development process, functional decomposition, object-oriented analysis, structured analysis, TOGAF ADM*

SYSTEM ARCHITECTURE

Approaches to Architecture Development

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand how to tailor and apply approaches, processes, and methodologies to develop architectures that support decision making. They should understand the scope, methodology, strengths, and weaknesses of various approaches so they can apply them, separately and in combination, to architecture development efforts.

Introduction

Multiple complementary approaches and methodologies are used to develop enterprise and system architectures. Some of the most popular approaches used in government departments and agencies are:

- U.S. Department of Defense Architecture Framework (DoDAF)
- The Open Group Architecture Framework (TOGAF)
- Object-oriented with Unified Modeling Language
- Spewak architecture process and Zachman Framework

The key steps of any architecture development approach are:

- Define the architecture purpose, value, and decisions it will support.
- Get information needed to define the architecture from stakeholders as early as possible.
- Create, refine, and update the architecture in an iterative way throughout the acquisition life cycle.
- Validate that the architecture will meet expectations when implemented.
- Define roles for team members to guide and coordinate their efforts.
- Create estimates and schedules based on the architectural blueprint.
- Use the architecture to gain insight into project performance.
- Establish a lightweight, scalable, tailorable, repeatable process framework [1].

Determining the Right Process/Method

Many SEs believe there is an “either-or” decision to be made regarding different architectural frameworks (e.g., DoDAF or TOGAF), but this is not necessarily the case. Some architectural standards address completely different elements of the architecting process; thus there may be a natural synergy among the frameworks. For example, TOGAF has a primary focus on architecture methodology—the “how to” aspect of architecting, without prescribing architecture description constructs. DoDAF has a primary focus on architecture description via a set of viewpoints, without a detailed specification of methodology [2].

DoDAF 6–Step Architecture Process

The primary focus of DoDAF is architecture description—the architecture depiction consisting of several models (called products in DoDAF-2004). Initially the primary objective of DoDAF was to facilitate interoperability among DoD systems; however, that objective has been broadened to assist decision making by DoD managers at all levels on issues relating to DOTMLPF—Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities—and DoD information technology systems.

Although a 6-step architecture process (see Figure 1) is described, it is meant to remain simple, tailorable, and able to be augmented by other architecture development processes.

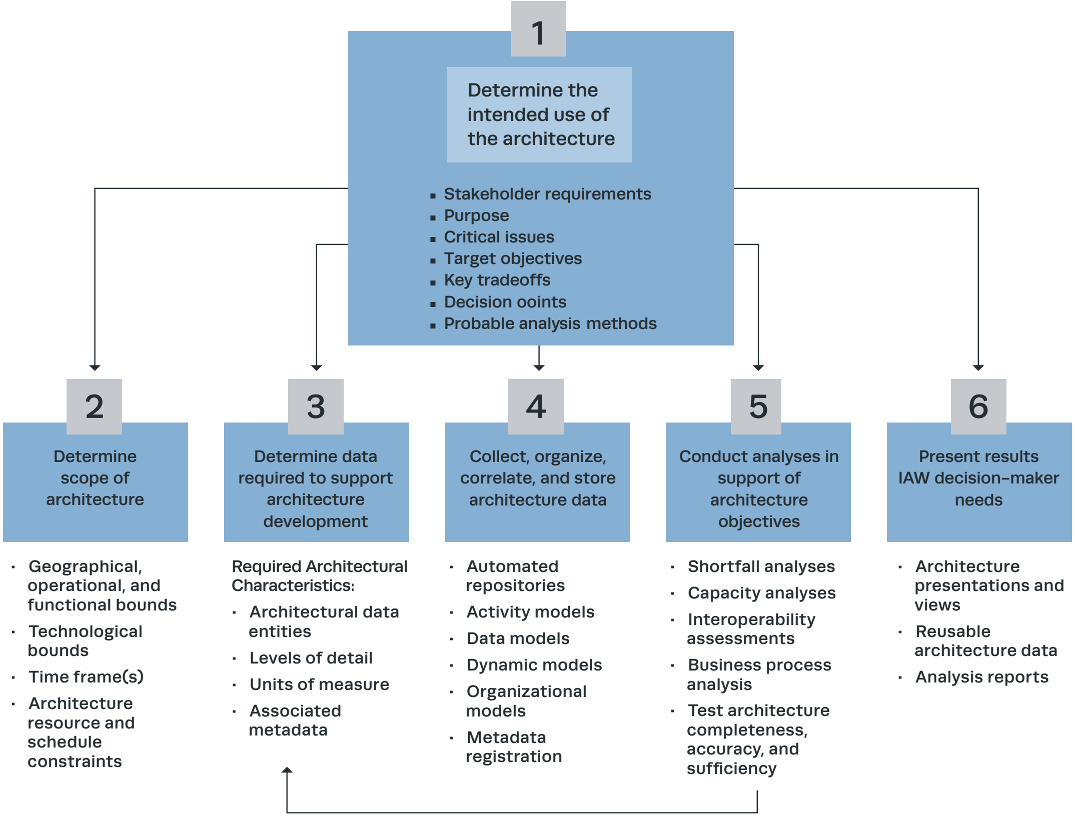

1
Determine the intended use of the architecture
Stakeholder requirements
Purpose
Critical issues
Target objectives
Key tradeoffs
Decision ooints
Probable analysis methods
2
Determine scope of architecture
Geographical, operational, and functional bounds
Technological bounds
Time frame(s)
Architecture resource and schedule constraints
3
Determine data required to support architecture development
Required Architectural Characteristics:
Architectural data entities
Levels of detail
Units of measure
Associated metadata
4
Collect, organize, correlate, and store architecture data
Automated repositories
Activity models
Data models
Dynamic models
Organizational models
Metadata registration
5
Conduct analyses in support of architecture objectives
Shortfall analyses
Capacity analyses
Interoperability assessments
Business process analysis
Test architecture completeness, accuracy, and sufficiency
6
Present results IAW decision-maker needs
Architecture presentations and views
Reusable architecture data
Analysis reports

Figure 1. 6-Step Architecture Process

The method described within DoDAF is generic and can be used with other frameworks. The process supports both the structured analysis and object-oriented analysis and design modeling techniques and their specific notations [3].

TOGAF Architecture Development Method (ADM)

The TOGAF Architecture Development Method (ADM) provides a tested and repeatable process for developing architectures. It is a generic method for architecture development that is designed to deal with most systems. However, it will often be necessary to modify or extend the ADM to suit specific needs. One of the tasks before applying the ADM is to review its components for applicability, and then tailor them as appropriate.

The phases within the ADM are as follows:

- **The Preliminary Phase:** describes the preparation and initiation activities required to prepare to meet the operational directive for a new architecture, including the definition of an organization-specific architecture framework and the definition of principles.
- **Phase A–Architecture Vision:** describes the initial phase of an architecture development cycle. It includes information about defining the scope, identifying the stakeholders, creating the architecture vision, and obtaining approvals.
- **Phase B–Business Architecture:** describes the development of a business architecture to support an agreed architecture vision.
- **Phase C–Information Systems Architectures:** describes the development of information systems architectures for an architecture project, including the development of data and application architectures.

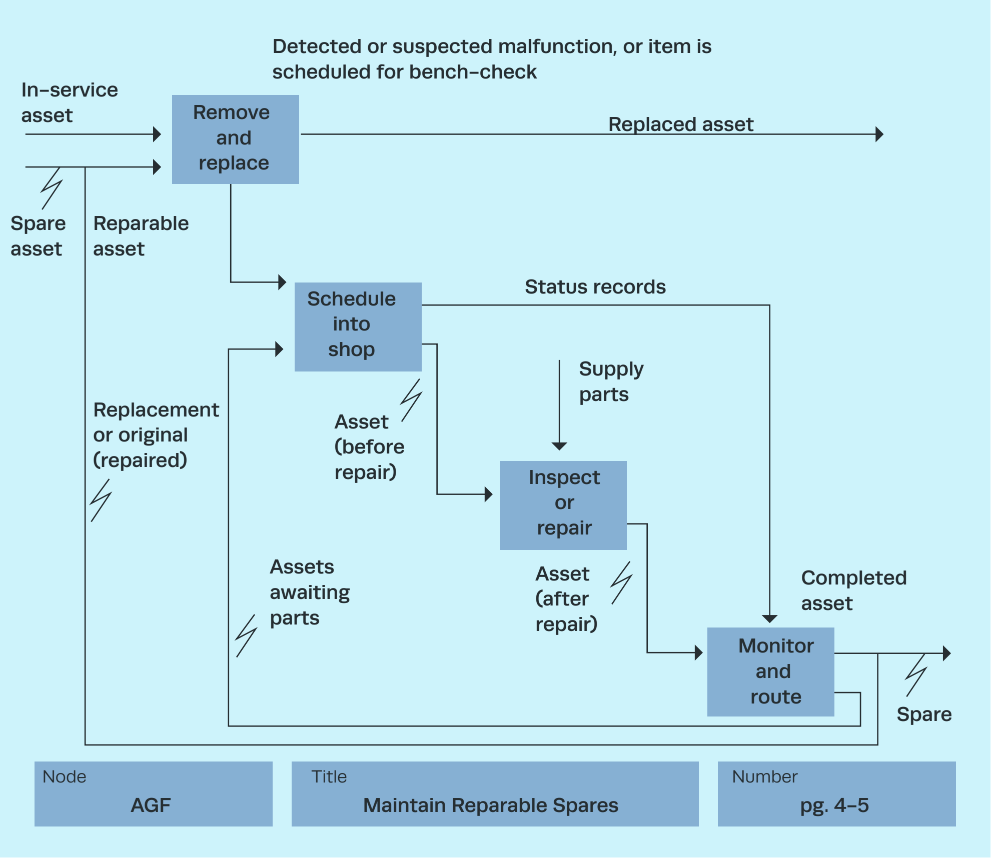


Figure 2. Integrated Definition for Function Modeling (IDEF0)

- **Phase D–Technology Architecture:** describes the development of the technology architecture for an architecture project.
- **Phase E–Opportunities and Solutions:** conducts initial implementation planning and identifies delivery vehicles for the architecture defined in the previous phases.
- **Phase F–Migration Planning:** addresses the formulation of a set of detailed sequences of transition architectures with a supporting implementation and migration plan.
- **Phase G–Implementation Governance:** provides an architectural oversight of the implementation.
- **Phase H–Architecture Change Management:** establishes procedures for managing change to the new architecture.
- **Requirements Management:** examines the process of managing architecture requirements throughout the ADM [4].

As a generic method, the ADM may be used in conjunction with the set of deliverables of another framework where these have been deemed to be more appropriate (e.g., DoDAF models).

Modeling Techniques

Once the decision about the architecture development methodology is resolved, selecting a technique to discover the architectural structure and processes is important. Currently two approaches are in use—object-oriented analysis and design, and structured analysis and design. Both have strengths and weaknesses that make them suitable for different classes of problems; however, the object-oriented methodology is better for complex, interactive, and changing systems with many interfaces, which are the kinds of systems most MITRE SEs face.

Structured functional techniques tend to be rigid and verbose, and they do not address commonality. Functional decomposition does not lend itself well to cases of highly complex interactive problems and is generally not used in modern development environments. Functional solutions are often difficult to integrate horizontally and are costly to sustain.

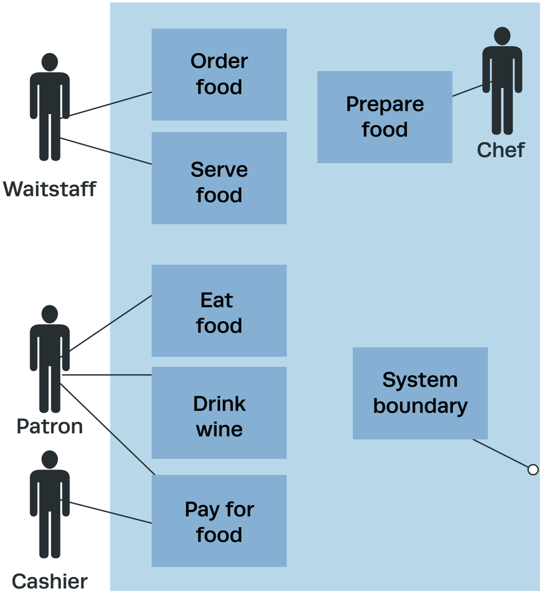


Figure 3. Use Case Diagram

The object-oriented method takes a value-based approach to discovering system capabilities. Use cases describe the behavior between the system and its environment (see Figure 3). From the use case, the services the system must provide are derived. Those services are then realized by the internal structure of the system elements in iterative steps until system elements are simple enough to build. The resultant set of diagrams traces the composition of the system from its parts to the aggregated behavior captured within the set of use cases.

Object-oriented approaches focus on interaction from the beginning, which has the beneficial side-effect of defining the boundary between the system and its environment. The use cases identify the ways in which the operator will use the system. Sequence diagrams (see Figure 4) illustrate the interactions the system must support. The "lifelines" of the diagram gather the behavioral responsibilities of each "object" participating in the use case. These responsibilities are the requirements to share data across the collection to produce the required result.

The advantages of the object-oriented method are that it embraces the concept of effects-driven process development, and it promotes reuse, facilitating the federation of cross-functional domain architectures. The focus on system interfaces also supports the service-oriented architecture implementation pattern [5].

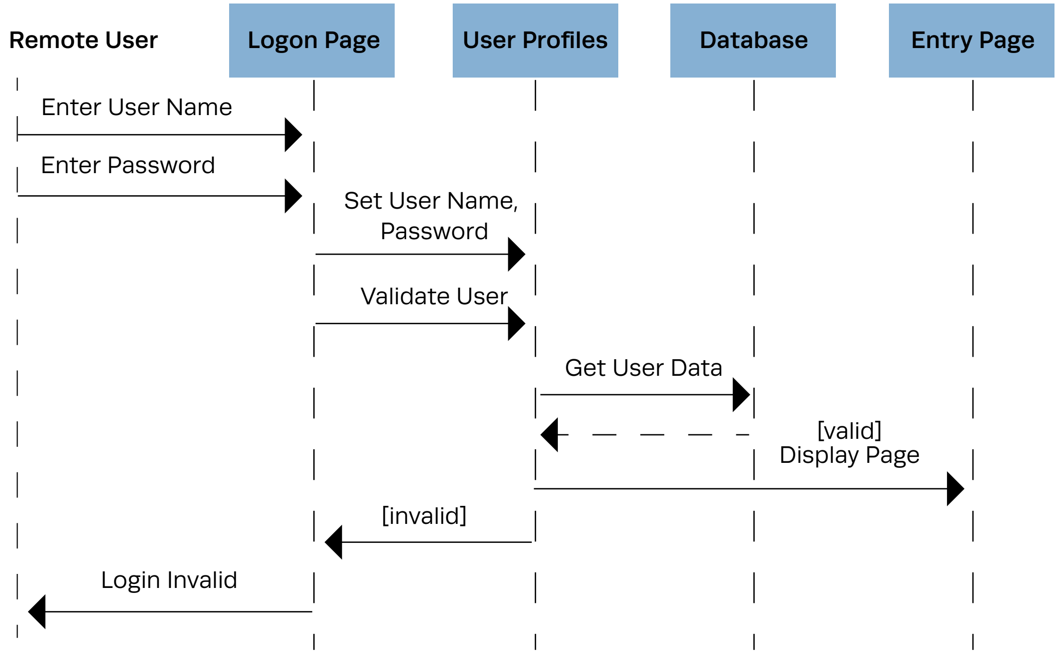


Figure 4. Sequence Diagram

For an example of a functional specification using use cases and sequence diagrams, see the *Functional Specification for Consolidated Afloat Network and Enterprise Services* [6].

Best Practices and Lessons Learned

Purpose before architecture. Purpose must drive the architecting effort or the effort will be subject to the criticism of architecting for its own sake.

Architecting is integral to systems engineering. Significant analytical insight into the system is gained through the process of architecting.

Think "both-and." Various architecture methodologies and approaches exist. When properly understood, they can be complementary. Some approaches and frameworks address architecture content, and others address the architecture process. Understand the value of all to apply the best course of actions for the purpose. Actively consider mixing and matching them to achieve your purpose.

Different models for different situations. Basic modeling techniques include a structured approach and an object approach. Understand their application strengths and weaknesses. The object approach provides many features to support complex system architectures and their interactions.

References and Resources

1. Lattanze, A., 2005, "The Architecture Centric Development Method," Carnegie Mellon University report CMU-ISRI-05-103.
2. Blevins, T., F. Dandashi, and M. Tolbert, 2010, TOGAF ADM and DoDAF Models, The Open Group White Paper.
3. U.S. Department of Defense, 2009, DoD Architecture Framework (DoDAF) Ver. 2.0.
4. The Open Group, 2009, The Open Group Architecture Framework (TOGAF).
5. Folk, T., 2006, Architectural Structures and Specifications.
6. Navy PMW 160 Tactical Networks, May 20, 2009, "Architecture Specification for Consolidated Afloat Network and Enterprise Services (CANES), Increment 1."

Additional References and Resources

Draft Federal Information Processing Standards Publication 183, December 1993 (withdrawn September 2008), Integration Definition for Function Modeling (IDEF0).

Federal CIO Council, 2001, A Practical Guide to Federal Enterprise Architecture.

Definition: *Architectural patterns are a method of arranging blocks of functionality to address a need. Patterns can be used at the software, system, or enterprise levels. Good pattern expressions tell you how to use them, and when, why, and what trade-offs to make in doing so. Patterns can be characterized according to the type of solution they are addressing (e.g., structural or behavioral).*

Keywords: *architecture, architecture patterns, patterns*

SYSTEM ARCHITECTURE

Architectural Patterns

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are frequently the stewards of an enterprise, system, or software architecture over its life cycle. The MITRE SE is expected to understand how architecture patterns can simplify and expedite the development of the system, and to mandate and encourage their use when appropriate.

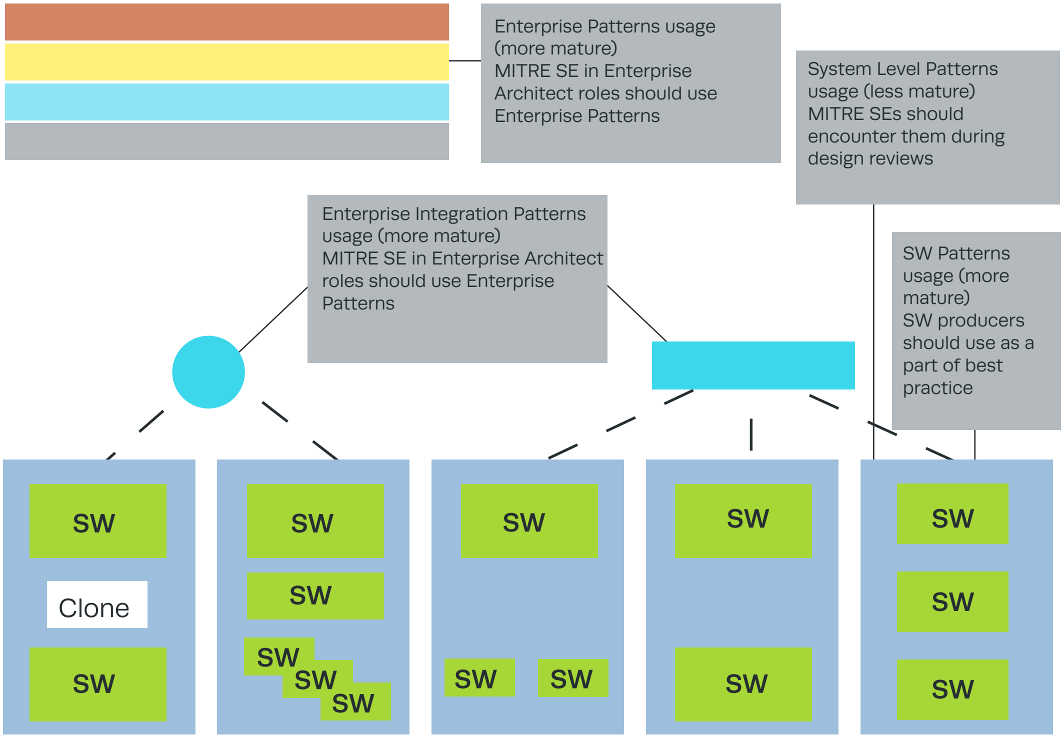

Enterprise Patterns usage (more mature)
MITRE SE in Enterprise Architect roles should use Enterprise Patterns
System Level Patterns usage (less mature)
MITRE SEs should encounter them during design reviews
Enterprise Integration Patterns usage (more mature)
MITRE SE in Enterprise Architect roles should use Enterprise Patterns
SW Patterns usage (more mature)
SW producers should use as a part of best practice
SW
Clone
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW

Figure 1. Architectural Pattern Usage

Background

"A key aspect to enterprise architecting is the reuse of knowledge. In most organizations today, the experience gained while doing a similar endeavor in the past is rarely utilized, or grossly underutilized, while dealing with a need today. Through better utilization of experiences and knowledge from the past, one can obtain major strategic advantages [1]." Pattern usage is an excellent way to reuse knowledge to address various problems. Figure 1 shows the levels of pattern application and how mature the pattern practice currently is for each one.

Definition

The architecture of an object, system, or enterprise is recognizable from the organization of features that contribute either structurally or behaviorally to the subject. A "pattern" has been defined as *"an idea that has been useful in one practical context and will probably be useful in others* [2, 3]."

Complexity Management

A major problem facing MITRE's sponsors today is constructing large, complex "systems of systems." These initiatives attempt to integrate dozens of legacy applications into a "system of pre-existing systems" to solve new and unexpected problems. The use of patterns can make these systems more efficient and effective. For instance, a system might have a tightly coupled architecture to address low-latency performance needs. Likewise, loosely coupled architectures may provide more opportunities to flexibly combine existing functions. As an example, one pattern used to enable loose coupling is the façade pattern in software architecture. This structural pattern provides a simple interface easily understood by many customers, hiding the complexity of function it provides, and is typically used when a service is to be provided to many objects in the environment.

Pattern Expression

One of the reasons why "experience gained while doing a similar endeavor in the past is rarely utilized" is because problems and their solutions are not expressed in a form suitable for reuse. Patterns provide a form for expressing technical solutions in the context of business problems and capturing them as reusable corporate knowledge assets.

In 1979, the (building) architect Christopher Alexander published *The Timeless Way of Building*, which describes a way to organize common solutions to architectural problems using patterns. In the early 1990s, software engineers began applying these ideas to systems architectures. Here is an example of a layered enterprise architecture expressed in Alexander's format:

- Name
  - *Layering*
- Context (situation giving rise to a problem)
  - Systems need to evolve to accommodate changing user requirements and new technologies
  - Managing change in complex systems
- Problem (set of forces repeatedly arising in the context)
  - Applications built as monolithic structures
  - Changing one part propagates costly changes everywhere
  - Migration timelines are long and expensive
- Solution (configuration to balance the forces)
  - Structure a system into layers
  - Each layer is a "black box" with well-defined interfaces
  - Implementation details of each layer are hidden behind the interface

Figure 2 illustrates the *Layering* pattern.

A pattern can be expressed using both human language such as prose, and more formal representations such as Unified Modeling Language diagrams. Patterns may also provide fragments of code to illustrate a design solution; however, it is not the intent of a pattern to provide a fully coded implementation.

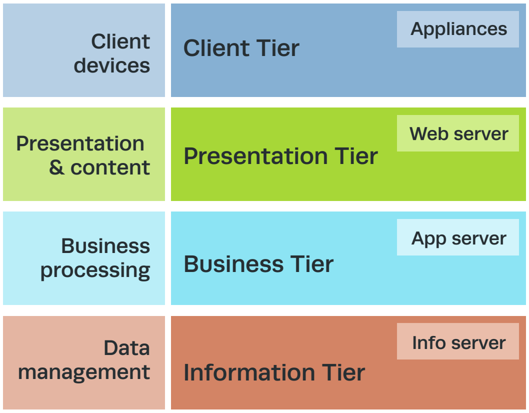


Figure 2. Layering Pattern

Applications of Patterns

As the value of patterns becomes recognized in the federal government, agencies are beginning to build pattern repositories in the context of the Federal Enterprise Architecture Framework. For example, the Department of Veterans Affairs has established a Technical Reference Model that includes 18 patterns that address such issues as router configurations and email address conventions.

When problem spaces are pervasive in an enterprise, there is an opportunity to develop guidelines in the form of patterns to address and govern solutions to that problem. The Tactical Edge Characterization Framework [4] contains patterns that address solutions to problems that occur at the edge of an enterprise where the users do not have large-scale and robust infrastructures.

The Navy has successfully applied patterns for their surface combat systems software product line. They use a layered presentation approach and a catalog of pattern elements.

Another set of problems occurs in the security domain of enterprises. The use of different approaches and a lack of patterns in developing security solutions lead to interoperability problems.

Best Practices and Lessons Learned

To be effective, patterns need to be incorporated into the corporate culture and adopted by management, business, and technical organizations. The effective use of patterns involves activities across technical, organizational, and process dimensions (see Figure 3).

In addition to internal corporate use, patterns can leverage collective solutions among partners across corporate, government, and national boundaries.

Seek out pattern sources. For systems you are the steward of, seek out sources of architectural patterns. Examples include Net-centric Enterprise

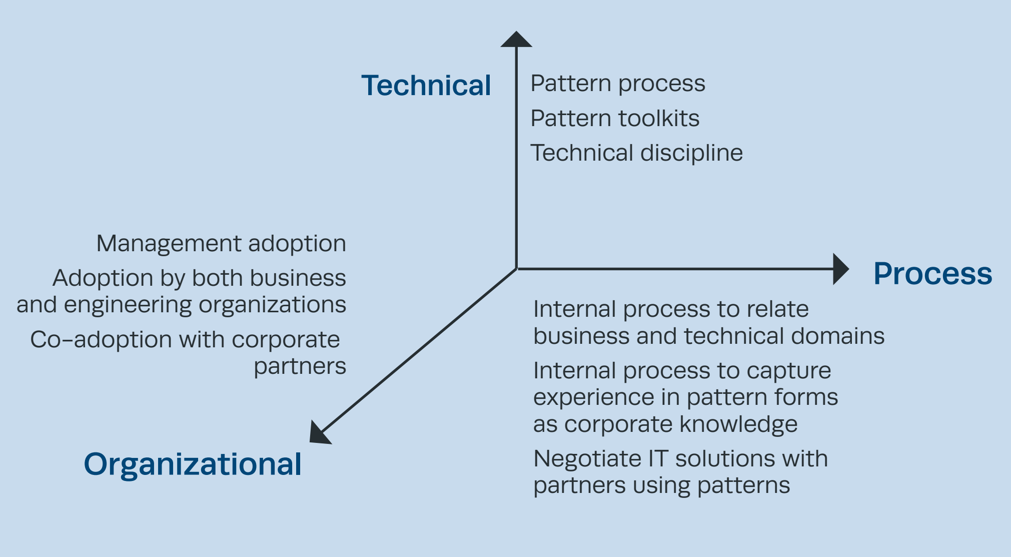

Technical
Pattern process
Pattern toolkits
Technical discipline
Process
Internal process to relate business and technical domains
Internal process to capture experience in pattern forms as corporate knowledge
Negotiate IT solutions with partners using patterns
Organizational
Management adoption
Adoption by both business and engineering organizations
Co-adoption with corporate partners

Figure 3. Dimensions of Effective Pattern Use

Solutions for Interoperability (NESI) [5], and the Electronic Systems Center Strategic Technical Plan [6]. These two are particularly applicable to problems of enterprise-level net-centricity.

Be a pattern steward. Recognize and capture patterns for reuse by others. Base patterns on proven experience in a common form or expression that is amenable to capture as a corporate knowledge asset. This is one way to help our customers solve their hard problems.

Lead the way in pattern usage. Enable and stimulate the selection of technical solutions based on successful patterns by using them in key documents such as Technical Requirements Documents, Software Development Plans, Systems Engineering Management Plans, and other key architecture documents.

Patterns, patterns, everywhere! Adopt patterns not only in technology-based SE work but also organizationally and in the process arenas. Works such as the Mission Level Modeling done by Prem Jain [7] contain workflow patterns that can be reused in architecture modeling efforts.

References and Resources

1. Peloquin, J. J., 2001, An Essay on Knowledge as a Corporate Asset: Building the Case for Human Capital.
2. Fowler, M., 1997, *Analysis Patterns—Reusable Object Models,* Addison-Wesley, ISBN 0-201-89542-0.

3. The Open Group, The Open Group Architecture Framework (TOGAF), ver. 8.1.1.
4. Dandashi, F., et al., November 2007, Tactical Edge Characterization Framework—Vol. 1: Common Vocabulary for Tactical Environments, The MITRE Corporation.
5. Department of the Navy, SPAWAR Systems Center Pacific, "NESI Public Site—Net-Centric Enterprise Solutions for Interoperability," accessed February 25, 2010.
6. "ESC Strategic Technical Plan," MITREpedia.
7. "Mission Level Modeling," MITREpedia.

Additional References and Resources

Adams, J., S. Koushik, G. Vasudeva, and G. Galambos, *Patterns for e-Business*, IBM Press, ISBN 1-931182-027.

Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture, A System of Patterns*, ISBN 0-471958-697.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, ISBN 0-201633-612.

Halley, M. R., and C. Bashioum, Enterprise Transformation to a Service Oriented Architecture: Successful Patterns, *Proceedings of the IEEE International Conference on Web Services (ICWS'05)*.

Hohpe, G., and B. Woolf, *Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions,* ISBN 0-321-20068-3.

Lapkin, A., October 22, 2004, *A User's Guide to Architectural Patterns*, Gartner Research Note G00123049.

Leganza, G., and J. Meyer, April 13, 2001, *Using Patterns in Enterprise Architecture: Part 1—Benefits and Drawbacks of the Patterns Methodology*, Giga Information Group.

MITRE, Information Sharing, Biopedia, accessed February 1, 2011 (includes instructions to access the Sharing Among International Partners Initiative [SAIPI] wiki).

Navy PEO Integrated Warfare Systems, July 31, 2009, *Surface Navy Combat Systems Architecture Description Document*.

Schulman, J., October 19, 2004, *Overcome Architecture Pattern Pitfalls and Problems*, Gartner Research Note G00123461.

Schulman, J., October 20, 2004, *Architecture Patterns Lead to Better Solutions*, Gartner Research Note G00123458.

The Open Group, The Open Group Architecture Framework (TOGAF), ver. 8.1.1, Part IV (Resource Base), Architecture Patterns.

System Design and Development

Definition: *System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.*

Keywords: *contractor, design, design review, development, evaluation, requirements, specifications, strawman, traceability, validation, verification*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to have a sound understanding of what a system requirement is intended to convey, what constitutes a good system requirement, how to identify a poorly written requirements statement, and what constitutes a good set of systems requirements. MITRE SEs are expected to be able to transform business/mission and operational needs into system requirements. Typically MITRE SEs lead the government acquisition program office effort to develop these requirements or are heavily involved in it. Collectively the descriptions and constraints comprising the system-level technical requirements are one of the most important products that MITRE can develop for the customer.

MITRE SEs are expected to help lead the government effort to create realistic top-level designs and associated risk mitigation activities so that planning will be based on a realistic foundation. Cost, schedule, and performance projections based on the top-level system design can be instrumental in mitigating and managing program risks. MITRE SEs are expected to be able to evaluate and influence the contractor's design and development effort, including making independent performance assessments and leading design review teams. In some programs, contractors will have primary responsibility for the top-level design with MITRE SEs providing guidance and verification of their efforts. In other programs, the government will develop a top-level design as part of its early systems engineering activities. Often MITRE will have a lead role or substantial responsibility for developing the government's top-level system design.

MITRE SEs are expected to understand the importance of system design in meeting the government's mission and goals. They are expected to be able to review and influence the contractor's preliminary design so that it meets the overall business or mission objectives of the sponsor, customer, and user. MITRE SEs are expected to be able to recommend changes to the contractor's design activities, artifacts, and deliverables to address performance shortfalls and advise the sponsor or customer if a performance shortfall would result in a capability that supports mission requirements, whether or not the design meets technical requirements. MITRE SEs are expected to be thought leaders in influencing decisions made in government design review teams and to appropriately involve specialty engineering.

Context

Core activities in system design and development include developing system-level technical requirements and top-level system designs and assessing the design's ability to meet the system requirements.

System-level technical requirements describe the users' needs, and provide information for the finished system to meet legal restrictions, adhere to regulations, and interoperate or integrate effectively with other systems. The system-level technical requirements are used by the government to acquire a capability, system, or product to meet a user need. They are used as part of a procurement contract solicitation or prototyping/experimentation effort and by the product vendors as their design criteria. The decisions made when defining system-level technical requirements can affect the number of potential solutions, the technical maturity of the potential solutions, system cost, system evolution, and development time and phasing.

System-level technical requirements are a critical precursor to and foundation of system design and development. A top-level system design is generally under the stewardship of the government team and represents the government team's independent projection of the way a system could be implemented to meet requirements with acceptable risk. The primary reason for developing a top-level system design is to provide a technical foundation for planning the program. It is the government's de facto technical approach to meeting the customer's needs. A top-level system design developed early in an acquisition program can be used to assess system feasibility and provide some assurance that the implemented design will satisfy system requirements. Done early in a program, a government design effort can be a powerful basis for developing fact-based government projections of cost, schedule, performance, and risk and provide the foundation for subsequent contractor design efforts.

Requirements traceability is a critical activity during the design, development, and deployment of capability that starts with the translation of the users' operational needs into technical requirements and extends throughout the entire system life cycle. It is a technique to develop a meaningful assessment of whether the solution delivered fulfills the operational need. Traceability is also the foundation for the change process within a project or program. Without the ability to trace requirements from end to end, the impact of changes cannot be effectively evaluated. In addition, change should be evaluated in the context of the end-to-end impact on other requirements and overall performance (e.g., see the SEG's Enterprise Engineering section). This bi-directional flow of requirements must be managed carefully throughout a project/program and be accompanied by a well-managed requirements baseline.

The articles in this topic highlight important elements of system design and development. The article "Develop System-Level Technical Requirements" provides guidance on selecting the right level of detail in writing technical requirements, highlights common challenges in achieving stakeholder agreement on requirements, suggests ways to handle them, and provides a checklist to help ensure that all bases have been covered in developing the system-level requirements. The article "Develop Top-Level System Design" provides guidance on early design efforts. The article is written from the perspective of an in-house government design activity, but many of the best practices and lessons learned can be used to shape, guide, and monitor contractor design efforts. The article "Assess the Design's Ability to Meet the System Requirements" provides guidance in establishing and accomplishing traceability, the importance of two-way traceability (both up and down), the need to have testing in mind when beginning traceability efforts, and the value of engaging with the operational user. It describes the technique of using a requirements traceability matrix to manage the specific traceability and verification from user need to requirements to design and development modules to test cases and measures/metrics for success.

For related information, see the SEG's Concept Development, Requirements Engineering, and System Architecture topics.

References

Wikipedia contributors, “Systems Design,” Wikipedia, accessed December 4, 2009.

Wikipedia contributors, “Systems Development Life Cycle,” Wikipedia, accessed December 4, 2009.

Definition: *System-level technical requirements is a general term used to describe the set of statements that identifies a system's functions, characteristics, or constraints.*

Keywords: *acquisition development program, requirement, specification*

SYSTEM DESIGN AND DEVELOPMENT

Develop System-Level Technical Requirements

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to have a sound understanding of what a system requirement is intended to convey, what constitutes a good system requirement, how to identify a poorly written requirements statement, and what constitutes a good set of systems requirements. MITRE SEs are expected to be able to transform business/mission and operational needs into system requirements, including [1]:

- Exploring and recommending creative ways to elicit, analyze, and document user requirements
- Transforming and integrating business/mission and operational needs into system requirements, including unstated or implied needs
- Promoting shared understanding and facilitating stakeholder agreement about systems requirements

- Integrating new requirements generated by prototypes into the system requirements
- Analyzing the interrelationships, priorities, cost, implementation, and environmental implications of system requirements
- Defining system boundaries, including how the system interacts with both inputs from and outputs to users, equipment, or other systems.

Background

Frequently MITRE plays a central role in developing system-level technical requirements, interpreting them, and assessing designs against them. Developing the right system requires the right information to communicate what the system is intended to do and what conditions or constraints its design must accommodate.

Collectively the descriptions and constraints that make up the system-level technical requirements are one of the most important products that MITRE can develop for the customer. Often, the system-level technical requirements are used in a government activity with the objective of acquiring a capability, system, or product to meet a user need. They are used as part of a procurement contract solicitation or prototyping/experimentation effort and by the product vendors as their design criteria. System-level technical requirements describe the users' needs, and provide information for the finished system to meet legal restrictions, adhere to regulations, and interoperate or integrate effectively with other systems. The decisions that are made when defining system-level technical requirements can affect the number of potential solutions, the technical maturity of the potential solutions, system cost, system evolution, and development time and phasing. Requirements definition (see the SEG's Requirements Engineering topic) confers an obligation and an opportunity for MITRE to influence the system development in ways that improve the ability of systems to interoperate with other systems. Poor requirements often result in solutions that are at high risk of failing to meet the user need, cost, schedule, and ability to interoperate.

Application

Ideally system-level technical requirements are developed after user requirements are defined. When not the case, a preliminary version of the system requirements may be drafted for developing a prototype or experimentation system to confirm, clarify, or discover user requirements. Prototypes or experimentation systems are also instrumental in validating that key technologies needed to meet the requirements are sufficiently mature and can meet the requirements that exist. In most government departments and agencies, system requirements are needed before development, manufacture, or construction. Once established, the system requirements exist and evolve for the life of the system. They may be and frequently are updated as the user defines new needs, or when the environment in which the system operates changes [2, 3]. In evolutionary or incremental acquisitions, the system requirements generally get defined in greater detail as the increment in which they are implemented draws near.

Many projects capture their system requirements using formal "shall" requirement statements in a system specification document. Use of the term "shall" in acquisition programs drives a mandatory implementation by the developer of a capability to meet the "shall." Selective use of "shalls" can be beneficial in situations in which needs are likely to evolve over time. Exhaustive and excessive use of "shall" statements can be constricting and costly. A term such as "should" can be used to both show government preference and allow some freedom in design and management/negotiation (e.g., time-driven, capability-driven, cost-driven) of system requirements intended to meet user needs over time.

The term "document" or "documented" has a long history in referring to requirements and is commonly used to this day. They should not be construed generally to mean that the information exists in paper form or is even organized as a paper substitute, such as a word-processed document [4]. Many projects today use electronic software tools to capture and manage requirements and other information. MITRE SEs are expected to understand the various types of tools and media, and their advantages and disadvantages, when capturing system requirements in different situations. Some projects, especially those involving software-intensive systems, use modeling tools or languages to describe major system elements, required behaviors, and interfaces. Examples include Unified Modeling Language (UML) and System Modeling Language (SysML).

Development of System-Level Requirements

When developing systems requirements, a good rule of thumb is to provide designers and test engineers with what they must know, but leave as much "white space" as possible for clever designers to explore design options (many times through prototypes of different forms or experiments).

An obvious place to start developing system requirements is with the user requirements—high-level expressions of user needs that the system is expected to satisfy. Examples of Department of Defense (DoD) user requirement sources include the initial capabilities document (ICD), the capability development document (CDD), and the capability production document (CPD).

There are two cautions when working with user requirement documents. First, the needs are frequently expressed in user operational lingo that may not be meaningful to engineers. An even more insidious problem is that while the language may be meaningful to both operators and engineers, it may also convey different interpretations, resulting in a lack of clarity about intent by the operators and the engineers. SEs need to explicitly account for and resolve this language divide and may have to translate operational terminology into engineering requirements. Second, user requirements often are not expressed in ways that unambiguously cover acceptance or test criteria. What appears to be a clear user requirement (e.g., "detect airborne objects out to a range of 100 miles") often requires the SE to do substantial work to achieve a set of requirements that are testable and can be built with reasonable technological risk.

In the requirement example above, there appears to be a straightforward performance requirement of 100 miles, but to a radar engineer it begs the question of "detect what?" and "how well?" There are both physics-related and technological challenges with detecting objects that are small and far away. Even for large objects, it is impossible to guarantee detection 100 percent of the time. To write an effective system requirement for a designer to implement a solution, an SE would have to derive additional requirements and/or verification criteria to define how small an object must be to be detected, and what the likelihood is that a given object within range will be detected.

Similarly, in a Maglev train development, the user's need to transit a great distance in a short time will eventually establish requirements for train speed parameters. The need to transport passengers will form the basis for safety requirements and maximum noise tolerances. In the vast majority of cases, the MITRE SE will benefit from working with the end users to create a mutual understanding of their needs and the capabilities and technology available and feasible to support them.

Consider using the checklist in Table 1 when developing a set of system-level requirements [4, 5, 6]:

Table 1. System-Level Requirements Checklist

Checklist Item	✔
The system-level technical requirements are traceable to the user requirements.	
Each system requirement describes something relevant: a function the system must perform, performance a function must provide, a constraint on the design, or a reference such as to an interface definition.	
The level of detail that the requirements provide about system functionality is appropriate. The requirements are sufficient to describe what the overall system must do, what its performance must be, and what constraints an engineer should consider. There are few requirements that specifically affect the design of only one component of the system. The major requirements drivers (e.g., those stressing the design) and associated risks should be identified.	

The requirements include any legal or regulatory constraints within which the system must perform. *Example: There may be restrictions on the use or quantity of certain hazardous materials in a system.*	
The requirements include enterprise architecture constraints within which the system must integrate (or toward which the system is desired to migrate). Requirements include appropriate open systems and modularity standards. *Examples: DoD Net-Ready requirements, modular open system architecture concepts, Electronic Systems Center strategic technical plan goals.*	
Environmental design requirements are specified. *Example: A control unit may be in a controlled office environment and the other major components may be outdoors, thus two environments must be defined and associated with the functionality operating in each environment.*	
All external interfaces for the system are included. Major internal interfaces may also be included if they are important to system modularity, or future growth in capability. These may include physical (mechanical fastening, electrical wiring, connectors), functional (mechanical stress transfer points, cooling, power sources, antennas, wire message formats, data exchanges), and software (software interface specifications, library calls, data formats, etc.). Remember that an internal interface between two subsystems that use a transport mechanism that is not part of the system is a hidden external interface. For example, two subsystems that communicate internally with each other over a sensitive but unclassified network as the internal interface (the data exchanged between them) and an external interface (the wiring and internet protocols to enable the data exchanges with the network).	
Requirement statements use the word "shall" or "should." The word "shall" has meaning in contractual language and is enforceable legally. Other words like "will," "may," "should," and "must" may show intent but are not legally binding in contracts. In some situations, it may be desirable to use "should" to show the government's intent and preference while at the same time allowing flexibility and latitude. *Example: "The system shall have a mean time between failures of greater than 500 hours."*	
Requirements statements are unambiguous. Terminology is clear without the use of informal jargon. Statements are short and concise.	

Performance requirements statements (including logistics/sustainment/support) are quantifiable, testable, and/or verifiable. Avoid the phrase "shall not." It is very difficult to prove a negative. Avoid qualitative words like "maximize" or "minimize." They force an engineer to judge when the design is good enough. The user may think that the engineer did not "minimize enough" and get into a legal argument with the contractor. *Note:* Every user requirements document includes: "the system shall be easy to use" requirement. Talk to other MITRE staff for examples from other projects and seek out a human factors specialist for requirements wording that is suitable both for specifying these requirements and methodologies for verifying them. Avoid specific, one-point values when defining requirements. Use ranges (minimum of, more than, less than, maximum of, within, etc.) to accommodate appropriate interpretation. Using a single point value may cause arguments if the system is tested at that exact value only, or if a test appears to be successful from an intent perspective, but does not meet the exact value stated in the system requirement. *Example: The system shall process a minimum of 100 transactions/sec.* *Example: The system shall be operable up to and including 30,000 ft.* *Example: The system shall operate in temperatures between 5 and 35 degrees Celsius.*	
If objective performance values are included as goals, ensure they are clearly identified and distinguished from firm requirements. User requirement documents refer to threshold requirements (those that must be provided), and objective requirements (better performance has value to the user, but not above the objective requirement). *Example: The system shall detect and display up to 100 targets within the surveillance volume with a goal of detecting and displaying up to 125 targets.*	
The operational and support environment is described and defined. *Example: The system shall be maintainable by an Air Force level 5 technician.* *Example: The system shall be reparable while in flight.*	
The requirements include appropriate use of Government and industry specifications, standards, and guides. Only include them if they are relevant and ensure that the correct version is listed in a list of reference documents.	
Verification approaches for all system performance and sustainability requirements are complete and appropriate. Every requirement must have a verification method identified. If a requirement cannot easily be verified by a direct inspection, measurement, or one-time demonstration of the requirement, the verification requirement should include an expanded test criteria description to ensure that there is no disagreement later in the program. This can include describing the number of trials, statistical criteria to be used, conditions of the test such as simulated inputs, etc.	

System requirements should be tracked or traced to the user requirements. Tracing a requirement means to cross-reference the source (in this case user) requirement on which a system-level requirement is based, and also to reverse reference which system requirement(s) implement the source requirements. Tracing user to system-level requirements helps ensure that all requirements have some user basis and that all user requirements are included in the system requirements for development. It is advisable to place the assumptions, constraints, and analyses associated with any derived requirements into a decision and/or requirements database as well.

It is possible to manually manage user to system-level requirement cross-references. Many projects use spreadsheets, databases, or word processors to manage requirement information. However, it is recommended that a project adopt a commercial requirements tool to aid in the process. Specialized database tools, such as DOORS (by IBM Rational), can be used to capture text requirements statements, diagrams, verification requirements, and other electronic files.

For related information, see the article “Assess the Design’s Ability to Meet the System Requirements.”

Best Practices and Lessons Learned

The devil’s in the (right level of) details. The primary challenge associated with developing system-level requirements is to provide enough detail so that there is sufficient information to implement the right system, yet not too much detail to restrict designers unnecessarily. One can think of a car analogy. If you were specifying requirements for a public transportation vehicle, you might specify the number of passengers, speed the vehicle must be capable of, and distance that must be covered. If it were intended to be a concept like an automobile, one would add requirements associated with single-user operation and with the regulations that affect automobile design to allow it to operate on public roadways.

With too high of a level of detail, the designer has insufficient information to provide a system that will meet the user need, and the designer will have to guess what was intended. In the automobile example, failing to include a requirement for a minimum amount of cargo space could result in a design with insufficient cargo space, yet the system would be compliant with the requirements. Do not assume that another person’s view of a logical, detailed design choice will match yours.

Too low of a level of detail can artificially constrain the design. Additional requirements may prevent a designer from making choices that can provide innovative solutions. This problem is often encountered because people have a natural tendency to want something they have seen and would like to be included in the final product. As a design concept is explored, people on the customer/sponsor side discover design details and then try to ensure that the particular feature of interest is included in the requirements. In the car example, such a person might try to specify a particular high-end sound system with custom interfaces for their favorite player when all that is required is compatibility with commonly purchased media such as compact disks.

System cost is likely to *increase* from adding too many low-level requirements to the system specification. Every system requirement provided to a contractor has a cost, even if they had already planned to include it in their design. Every requirement must be documented, allocated to design specifications, tracked, and formally tested. Contractors will bid that work. In addition, the detail requirements become a permanent part of the top-level system requirements for all future upgrades until they are explicitly changed.

Close communications with the users and use of prototypes, experiments, etc., are mechanisms to ensure a collective understanding of what is needed and to manage level-of-details requirements issues.

Stakeholders' agreement. Because system-level technical requirements are central to the definition of what is to be procured and have significant effects on program acquisition risk, cost, and schedule, reaching agreement on system requirements is challenging but critical. Many stakeholder organizations and individuals often have differing opinions. Contractors may argue to mitigate their risk or slant a requirement in their favor. Users may try to guide design or get additional capability. Co-workers and the sponsor have opinions based on their experience and perceptions. Sometimes political implications (real or assumed) exist when a requirement challenge is initiated by a sponsor senior leader or end user. If you are an engineer defining system technical requirements, you will eventually encounter arguments about:

- What a requirement statement or a specific word within a requirement statement means
- Whether a requirement should be included or excluded
- Whether a requirement is specifying too much or too little performance
- Whether a requirement directs design and should be removed
- Whether a requirement should be added to guide design and ensure that a specific feature is provided.

To resolve requirement arguments, it is often helpful to understand the source of the objection to your system requirement. The source of someone's argument often comes from one of the following situations:

- They have experience and are really trying to help.
  - They might not always be right, but they believe they are helping build a better set of requirements.
  - You may have to do more research or make a judgment as whether their position has merit.
- They have a concern about how someone else may interpret the requirement.
  - *Contractors:* They are afraid of how their words could be interpreted by others. Losing an argument about an

interpretation of a requirement late in the program is highly undesirable to the contractor because of the higher cost and schedule impact due to late design changes.

- *Program Office person or user representative*: They are afraid the requirement will not force the contractor to provide a specific design feature that they want included in the system.

- They want the program to adopt a competing technical approach.
  - *Contractors:* They want to slant the acquisition toward their specific solution to get the contract award or allow them to meet the requirement with a solution they have already identified.
  - *Any Party:* They may have a valid technical solution or they may want to have the project adopt their requirement over yours to demonstrate their contribution.
- They insist on a specific detail or feature the system must have, and they want specific words to include as a requirement. Any party can fear that if you don't specify the requirement explicitly, they will not get it.

Resolving any of these issues requires a mixture of negotiation ability, technical expertise, and an ability to understand the root cause of the other person's concern. Choosing clear requirements language is a good starting point. Being consistent with specification wording and using terminology similar to that employed in past projects with the contractor can also help. Understanding and experiencing the operational environment will give the MITRE SE additional knowledge on which to judge the requirements. In other cases, the SE will have to explore the other person's objections and determine whether their position has merit, technical or otherwise.

For related information, see the SEG topics Requirements Engineering and System Architecture.

References and Resources

1. The MITRE Corporation, "Requirements Engineering," MITRE Systems Engineering Competency Model, section 2.2.
2. Department of Defense, December 8, 2008, DoD Instruction 5000.02.
3. Chairman of the Joint Chiefs of Staff, March 1, 2009, Joint Capabilities Integration and Development System (CJCSI 3170.01G).
4. Stevens, R., P. Book, K. Jackson, and S. Arnold, 1998, *Systems Engineering: Coping with Complexity*, Prentice Hall.
5. Blanchard, B., and W. Fabrycky, 1998, *Systems Engineering and Analysis*, Prentice Hall.
6. International Council on Systems Engineering (INCOSE), January 2010, INCOSE Systems Engineering Handbook, Ver. 3.2, INCOSE-TP-2003-002-03.2.

Additional References and Resources

Navy PEO Integrated Warfare Systems, October 27, 2008, *Surface Navy Combat Systems Development Strategy Acquisition Management Plan (AMP).*

Definition: *In acquisition-oriented systems engineering, a top-level system design represents the envisioned implementation of a system in sufficient detail to support credible projections of cost, schedule, performance, evolution, and risk. It helps in assessing system feasibility at the program's outset, performing analyses of alternatives, and finalizing requirements and budgets prior to contract solicitation. If carefully developed, the design becomes the program's early technical baseline for acquisition planning activities.*

Keywords: *cost analysis, cost analysis requirements document, early design, early system design, early systems engineering, requirements optimization, technical baseline, top-level system design*

SYSTEM DESIGN AND DEVELOPMENT

Develop Top-Level System Design

MITRE SE Roles and Expectations: During initial capability planning activities, the MITRE systems engineer (SE) is often involved in establishing a sound program baseline, which includes an understanding of the system operational requirements, the system design concept, the architecture, the technical requirements, and the associated program cost and schedule. In these situations, a MITRE SE is expected to:

- Understand the purpose and role of top-level system design in the acquisition process
- Understand how and when a top-level system design should be undertaken
- Understand the associated benefits and risks
- Identify and engage subject matter experts (SMEs) with core technical skills appropriate for developing the top-level system design

- Apply the top-level system design baseline and lessons learned from the design activity in program acquisition planning activities.

Background

A top-level system design represents the government team's independent projection of the way a system could be implemented to meet the prevailing requirements with acceptable risk. Although a top-level system design could be mandated for eventual implementation by a development contractor, it is generally under the stewardship of the government team. The primary reason for developing a top-level system design is to provide a technical foundation for planning the program. It is the government's *de facto* technical approach to meeting the customer's needs. Once defined, a top-level system design represents an approach that can be used to develop a program schedule and cost estimates consistent with the program's technical content, as well as risk assessments, acquisition strategies, a logistics approach, etc. If a program is required to develop a cost analysis requirements document (CARD), the top-level design can be the foundation of that work.

Because the government is not in the business of designing systems, the top-level system design is only representative of what can be built with conventional technology; therefore, the design may sometimes indicate lower performance or greater cost than the design that eventually will be developed by the contractor, which may include proprietary innovations (particularly in a competitive environment). MITRE SEs are expected to help lead the government effort to create realistic top-level designs and associated risk mitigation activities so that planning will be based on a realistic foundation. Cost, schedule, and performance projections based on the top-level system design should include margins that will tend to help mitigate program risks.

As with many aspects of acquisition-oriented systems engineering, the role played by top-level system design varies considerably across programs. In some programs, any design—even a "top-level" one—is viewed as treading excessively into implementation territory, and is therefore considered the responsibility of the development contractor rather than the government. Even programs that do not share this philosophical view often do not have enough systems engineering resources for development of a top-level system design and the mandated architectural frameworks. The effort needs to be a teaming approach of the government, MITRE SEs, support or development contractors, and industry. In some cases, academia will also have a role in helping with ideas for technology and capabilities that could be included in a top-level design.

The process of developing a top-level system design engages the MITRE SE in the program technical content and provides numerous benefits:

- By analyzing the known/draft system operational requirements, the SE can discover the design implications of these requirements, identify any potential requirements conflicts or technically infeasible requirements, review and comment on the operational documents being developed, identify and possibly initiate requirements trades and/or risk reduction activities such as prototypes and experiments, and assess the affordability of the required system. This allows for smarter early interaction with the authors of the operational requirements document(s), producing clearer expectations and better defined operational documents (see the SEG's Requirements Engineering topic).
- By developing a top-level system design, the SE discovers the complexities, dependencies, and interactions within the system, gaining a better understanding of the program's technical concerns, issues, evolution needs, and risks that will have to be managed during program acquisition. The natural interplay between a top-level system design and system architecture should be captured in each (see the SEG's System Architecture topic).
- The development of the top-level system design requires the SE to explore industry's (and, at times, academia's) capabilities and the available technologies. This includes an assessment of interest by, and capability of, the potential contractors, and maturity and availability of required technologies. Early involvement with the potential contractors for the program tells the SE what questions to ask in the Request for Information (RFI) solicitation, enables intelligent discussions with industry during Industry Days, and results in a Request for Proposal (RFP) package that will be clearer, resulting in better quality proposals. A good understanding of the maturity (or lack thereof) of relevant technologies helps the SE define appropriate program acquisition and risk mitigation strategies. Communicating the users' needs and requirements together with the top-level design to industry partners is key in helping them understand the need and formulating the best solutions for the capabilities.
- The development of the top-level system design requires the SE to investigate availability, maturity, and applicability of products (systems, subsystems, components, algorithms, etc.) that could be used to provide some of the required system functionality and performance, thus getting an early assessment of the risk/benefit trade-offs associated with these products.
- The top-level system design process reviews the technical content and lessons learned from any precursor programs. The deployed capabilities, technologies, products, approaches, and solutions are available to the SE to consider for the new program.

The development of a top-level system design essentially mimics the first part of a system design and development program, stepping through abbreviated phases of requirements definition, decomposition, analysis, assessment of alternative solutions, and development of a design.

A top-level system design may be independent of technology, contractors, and existing systems. It needs to be detailed only enough to provide inputs into program cost and schedule estimation activities to allow for program planning activities (acquisition strategy, risk reduction, etc.). However, with many interfacing capability needs, use of services from repositories, and constraints of the technical environment where the capability must reside, a design cannot be completely agnostic to the technology and existing systems, and must consider the overall context and enterprise into which the capability will reside (see Enterprise Engineering section).

Finally, top-level system designing is a recurring activity. Initially, it can start as a refinement of the alternative selected from the analysis of alternatives; in time, it is updated and fleshed out as more information becomes available after the technology readiness assessment, RFI/Industry Days, user requirements working groups, completion of risk reduction activities, etc.

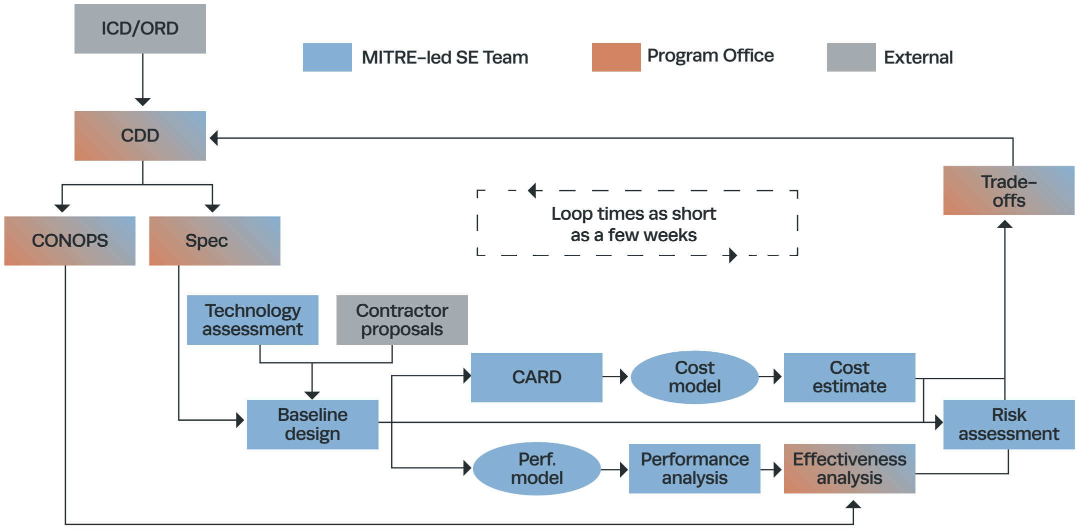


Figure 1. The Top-Level System Design Loop

The Top-Level System Design Process

Figure 1 shows how the top-level system design serves as the core of the program office's systems engineering analysis loop.

The figure represents a snapshot taken just prior to issuance of the RFP for the system development phase, at which point the major systems engineering challenge facing the program was the finalization of system requirements and the program budget. However, the same process was used earlier in the program to establish system feasibility, and identify and assess the maturity of critical technologies. Subsequently, after system acceptance, planning and budgeting for upgrades can be established.

Best Practices and Lessons Learned

Representative design. During the top-level system design activities, the SE has to be continually aware that the top-level system design may be just one of many possible solutions. The SE should keep an open mind to all available designs and ensure as "generic" a top-level system design as possible, one that will enable competition by all relevant industry members. Absent compelling performance reasons, the top-level system design solution should avoid locking in on a one-party approach (e.g., using an idea, solution, or existing product available to only some contractors).

Competing designs. Sometimes two mutually exclusive approaches are possible for the system design (e.g., software intensive vs. hardware solutions, or reuse of extensive existing products vs. new development). In this case, basing the program plans on a single top-level system design may have negative consequences because it could result in development of system requirements that would preclude the bid of one (or more) contractors, or result in unrealistic program cost/schedule estimates. In this instance, the SE should help evaluate the alternatives, identify and consider their pros and cons, and carefully decide which top-level system designs should be developed and carried forward into program planning to cover the range of implementation options.

Applicable, feasible, affordable, phased. Another challenge for the SE is to keep the top-level system design feasible and affordable and not fall into the trap of capturing the best features and all the additional capabilities of the various possible approaches, alternatives, technologies, and products. It is important to ensure the top-level system design meets the stated operational needs and is affordable and available in useful time. The implementation of solutions to complex problems is likely to be time-phased. Therefore, the top-level design should plan for evolutions of capabilities based on urgent user needs, technical feasibility, and affordability over time. Additional "bells and whistles" may be exciting, but they should be avoided because in the long run they risk breaking the program's bank.

Design depth. Developing a top-level system design is hard for SEs. By definition, the top-level system design is not intended to resolve all problems, address all issues, or mitigate all technical risk. Letting go of a partially completed design can be hard. The trick is to understand which parts of the top-level design need what degree of depth in order to address the critical uncertainties to be explored at this stage of the life cycle. That depends on an assessment of the key risks for the particular program and the relevant technologies.

An example of a comprehensive top-level system design comes from a government acquisition program that developed its capability within budget and ahead of schedule, earning the government-contractor program team accolades for exemplary acquisition excellence. MITRE serves as the lead systems engineering organization for the government program office team, which includes multiple systems engineering and technical assistance contractors, government laboratories, and other federally funded research and development centers.

One of the cornerstones of the program's systems engineering approach is the conviction that the best way for program offices (POs) to obtain reliable projections of cost, schedule, performance, and risk is through independent analysis. A comprehensive top-level system design, together with a government team with enough subject matter expertise to develop and use it, is the crucial enabler for such analysis.

Depth is key. To support meaningful assessments of feasibility, cost, and performance, a top-level system design must be detailed enough to identify and describe critical items. The level of detail needed to get to "critical items" depends on enabling technology, requirements, and the like. In this particular program, sufficient detail required identification and description of critical items such as enabling chips and components (e.g., low-noise amplifier monolithic microwave integrated circuits, high-power-density direct-current to direct-current converters, heat exchangers), key algorithms (e.g., false target mitigation), and all required computer software configuration items. The required depth was about the same as that of the technical volume of a system development proposal; in the case of the referenced program, the top-level system design document was more than 300 pages, and extended to work breakdown structure (WBS) Level 4—and in some cases, Level 5.

A typical objection against this degree of depth is that "nobody is ever going to build this government design anyway, so it's better to base programmatic projections on conceptual designs provided by the competing contractors." One answer to this objection is that costs and risks are largely determined by decisions made early in a program's life, well before any contractor-developed conceptual designs exist and often before candidate contractors have been identified. A more compelling answer is that, in a competitive environment, there are strong incentives for even the best intentioned contractors to underestimate the challenges associated with meeting the requirements. Conversely, in a sole-source environment, there may be strong incentives to overstate those challenges, particularly by a contractor who is already producing a system that could be viewed as a competitor to the envisioned system.

For example, in the referenced program the original operational requirements called for capabilities that in many ways exceeded those of any extant airborne system. At that time, other airborne sensors were tightly integrated into the host airframes, requiring extensive aircraft modifications. In contrast, to minimize cost and maximize operational availability, the operational user organization wanted the bulk of the system mission equipment to be encapsulated in a self-contained pod that could be rapidly moved between small—and lightly modified—host aircraft. At the time these requirements were floated, the industry consensus was that they were infeasible. However, the comprehensiveness and depth of the top-level system design provided the sponsor with the confidence to properly budget and proceed with the acquisition, which eventually proved highly successful.

Don't skimp on the team. The team required to develop a sound top-level system design of the requisite technical depth is substantial—about 30 to 40 people in the case of the referenced program—with total effort of about 30 staff-years. Of these, about 70 percent were SMEs, including hardware and software engineers, cost analysts, operational analysts, and experts in modeling and simulation. Approximately 20 percent were process experts, responsible for planning, configuration management, requirements traceability, and acquisition process compliance. Finally, 10 percent were technical leaders, including a chief engineer and system architect. Critical to all of this is involving the end-user community in the design process all along the way to help with trade-off analysis and negotiation of needs.

For this particular program, the same demographic makeup was also about right for virtually all of the other technically intensive government systems engineering activities necessary to support this acquisition, including proposal evaluation. For example, the top-level system design team eventually formed the core of the cost and technical boards that advised during proposal evaluations; their experience in developing and costing the top-level system design was invaluable in ensuring thorough proposal evaluations.

Involve cost analysts in the design synthesis. A key use for a top-level system design is to serve as the basis for cost projections; a properly documented top-level system design meets all the requirements for the DoD-mandated CARD.

The referenced program's experience shows there is substantial room for error in the subsequent cost estimate, no matter how thoroughly a design is documented, unless the cost analysts are involved in the design synthesis. Cost analysts can make substantive contributions to the design effort by helping to define an appropriate WBS, which effectively serves as the outline for the top-level system design document, and identifying the type of technical information that should be documented for each WBS area to facilitate costing.

An added benefit of involving the cost analysts up front is that cost models can be developed in parallel with the design synthesis, enabling the cost estimates to be provided sooner (about two months sooner, in the case of the referenced program) than those developed in a sequential process.

As previously mentioned, a crucial benefit to participating in the design process is that it gives the cost analysts greater insight into the salient technical issues, enabling them to perform a thorough evaluation of the cost proposals submitted by the contractors. Although difficult to quantify, this benefit was evident in the program's source selection.

MITRE's organic cost analysis capability and experience in tightly integrating that capability with extensive in-house technical subject matter expertise was unique among the many organizations supporting the acquisition program, and the resulting synergy that was achieved provided major added value to the program.

Involve performance modelers in the design synthesis. The top-level system design also serves as the basis for system performance projections. As with cost estimation, involving operational analysts and performance modelers up front allows the required models to be developed sooner, while improving the fidelity of the results.

Early development of the performance model provided a key benefit for the referenced program: it enabled the government to include the MITRE-developed model—but without the design parameters specific to the government design—as government-furnished information in the RFP packages provided to candidate offerors. Offerors were required to provide design-specific input files for the model as part of their proposals. This enabled the government to use a common tool to rapidly evaluate the performance of the competing designs—the same tool that was used to help set the requirements in the first place.

This continued to pay dividends beyond the source selection. The program's development contract called for the selected offeror to include a performance model as part of the delivered system (e.g., to support mission planning). The contractor was free to adapt an existing model for this purpose or to develop one from scratch. Because the MITRE/government model had been developed and already had been validated within the government team, the contractor elected to use it as the basis for the delivered model, thereby reducing cost and risk.

Organizationally collocate the key members of the team. Developing a sound top-level system design requires extensive face-to-face communication, necessitating that most team members be physically collocated. However, even with physical collocation, risks increase when the team is spread across multiple organizations and multiple management chains of command. The experience of the referenced program shows that synthesizing and leveraging a top-level system design becomes problematic unless key hardware SMEs, key software SMEs, and key cost analysts are all resident within the lead systems engineering organization. Although detailed-level design and implementation activities may sometimes be well accomplished by allocating tasks to hardware, software, and cost support organizations that are "best" in each discipline, early design work should be accomplished by a close-knit team representing all the required specialties, as well as overall systems engineering leadership.

Ensure the top-level system design is an independent government product. Contractor-developed conceptual designs may already exist when the government team undertakes development or update of a top-level system design. Because the top-level system design is used to help set requirements and budgets, inclusion of any of the contractors' proprietary innovations in the government design, thereby revealing an ostensibly preferred design approach, could destroy the integrity of a source selection. The government and MITRE must be cautious stewards of the top-level design. If a decision is made to share the design with industry, it must be shared with all interested industry partners equally. Even if care is taken to avoid inclusion of proprietary features, however, a top-level system design often resembles at least one of the contractor-developed conceptual designs. This should present no problems, provided that a paper trail can show the design was independently derived. For example, the referenced program's top-level system design provided an extensive design rationale that illustrated how the design flowed naturally from the requirements, based on technologies and sensor design practices documented in the open literature.

Definition: *The ability of a system design to meet operational, functional, and system requirements is needed to meet a system's ultimate goal of satisfying mission objective(s). One way to assess the design's ability to meet the system requirements is through "requirements traceability." This is the process of creating and understanding the bidirectional linkage between requirements (operational need), organizational goals, and solutions (performance).*

Keywords: *assessment, concept of operations (CONOPS), functional requirements, mission and needs, operational requirements, performance verification, requirements, requirements traceability, requirements traceability matrix, system requirements, traceability, verification*

SYSTEM DESIGN AND DEVELOPMENT

Assess the Design's Ability to Meet the System Requirements

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the importance of system design in meeting the government's mission and goals. They are expected to be able to review and influence the contractor's preliminary design so that it meets the overall business or mission objectives of the sponsor, customer, and user. MITRE SEs are expected to be able to recommend changes to the contractor's design activities, artifacts, and deliverables to address performance shortfalls and advise the sponsor or customer if a performance shortfall would result in a capability that supports mission requirements, whether or not the design meets technical requirements. They are expected to be thought leaders in influencing decisions made in government design review teams and to appropriately involve specialty engineering.

In requirements traceability and performance verification, MITRE SEs are expected to maintain an objective view of requirements and the linkage between the system end-state performance and the source requirements and to assist the government in fielding the best combination of technical solution, value, and operational effectiveness for a given capability.

Background

Traceability and Verification Process

A meaningful assessment of a design's ability to meet system requirements centers on the word "traceability." Traceability is needed to validate that the solution delivered fulfills the operational need. For example, if a ship is built to have a top speed of 32 knots, there must be a trail of requirements tied to performance verification that justifies the need for the additional engineering, construction, and sustainment to provide a speed of 32 knots. The continuum of requirements generation and traceability is one of the most important processes in the design, development, and deployment of capability.

Traceability is also the foundation for the change process within a project or program. Without the ability to trace requirements from end to end, the impact of changes cannot be effectively evaluated. Furthermore, change should be evaluated in the context of the end-to-end impact on other requirements and overall performance (e.g., see the SEG's Enterprise Engineering section). This bidirectional flow of requirements should be managed carefully throughout a project/program and be accompanied by a well-managed requirements baseline.

Requirements Flow

The planned functionality and capabilities offered by a system need to be tracked through various stages of requirements (operational, functional, and system) development and evolution. Requirements should support higher level organizational initiatives and goals. It may not be the project's role to trace requirements to larger agency goals. However, it can be a best practice in ensuring value to the government. In a funding-constrained environment, requirements traceability to both solutions as well as organizational goals is essential in order to make best use of development and sustainment resources.

As part of the requirements traceability process, a requirement should flow two ways: (1) toward larger and more expansive organizational goals, and (2) toward the solution designed to enable the desired capability.

Requirements Verification

Because the ability to test and verify is a key element of project/program success, requirements tied to operational needs should be generated from the outset and maintained throughout the requirements life cycle with test and verification in mind. Advice on the ability to test requirements can be extremely effective in helping a project or program. Techniques such as prototyping and experimentation can help assess requirements early and provide a valuable tool for subsequent verification and validation. For more information, see the article "Competitive Prototyping."

Test and verification plan development and execution should be tied directly back to the original requirements. This is how the effectiveness of the desired capability will be evaluated before a fielding decision. Continual interaction with the stakeholder community can help realize success. All test and verification efforts should relate directly to enabling the fielding of the required capability. Developing test plans that do not facilitate verification of required performance is an unnecessary drain on resources and should be avoided.

Before a system design phase is initiated, it should be ensured that the system requirements, captured in repositories or a system requirements document, can be mapped to functional requirements (e.g., in a functional requirements document [FRD]). The requirements in the FRD should be traceable to operational requirements (e.g., in an operational requirements document or a capabilities development document). If all this traceability is ensured, there is a better likelihood that the design of the system will meet the mission needs articulated in a concept of operations and/or the mission needs statement of the program.

Design Assessment Considerations

The design of a system should clearly point to system capabilities that meet each system requirement. This two-way traceability between design and system requirements will enable higher probability of a successful test outcome of each system requirement, the system as a whole, and delivery of a useful capability.

As the service-oriented architecture (SOA) approach matures, there is increased emphasis on linking system requirements to specific services. Therefore, the artifacts or components of system design should be packaged in a manner that supports provisioning of services offered within SOA (assuming that deployment of SOA is a goal of an enterprise).

For example, assume that the system requirements can be grouped in three categories: Ingest, Analysis, and Reporting. To meet system requirements within these categories, the design of the system needs to point to each component of the system in a manner that addresses how it would fulfill the system requirements for Ingest, Analysis, and Reporting, respectively. The design information should include schematics or other appropriate artifacts that show input, processing, and outputs of system components that collectively meet system requirements. Absent a clear roadmap that shows how input, processing, and outputs of a system component meet a given system requirement, there is risk in whether that specific system requirement will be met.

The Requirements Traceability Matrix: Where the Rubber Meets the Road

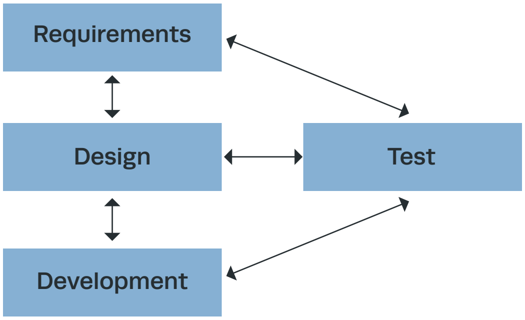


Figure 1. Traceability Matrix Relationships

Typically the project team develops a requirements traceability matrix (RTM) that shows linkage between functional requirements, system requirements, and system capabilities of system design components. An RTM that can clearly point to system components that are designed to meet system requirements is more likely to result in a well-designed system, all other considerations being equal. Additional linkages can be included in an RTM to show mechanisms to test functionality of system components for testing the design of the system to meet system requirements. An RTM as described above (i.e., one that ranges from statement of a requirement to methodology to test the system component that satisfies the system requirement) will go a long way in successfully assessing a system design to meet system requirements.

A traceability matrix is developed by linking requirements with the design components that satisfy them. As a result, tests are associated with the requirements on which they are based and the system is tested to meet the requirement. These relationships are shown in Figure 1.

A sustained interaction is needed among members of a requirements team and those of design and development teams across all phases of system design and its ultimate development and testing. This kind of dialog helps ensure that a system is being designed properly with an objective to meet system requirements. In this way, an RTM provides a useful mechanism to facilitate the much-needed interaction among project team members.

Table 1 is a sample RTM that spans “Requirement Reference” to “Design Reference.” The matrix can be extended to include testing mechanisms for further assurance that the system design will meet system requirements. The RTM in Table 1 links a system requirement to a design component (e.g., a name of a module).

The assessment of a system design should consider how well the design team presents the linkage of its design to system requirements (i.e., through documentation, presentations, and/or critical design reviews). A traceability matrix can be an important device in communicating a design’s ability to meet system requirements.

As part of the system design approach, the design team may develop mock-ups and/or prototypes for periodic presentation to the end users of the system and at design reviews. This

Table 1. Sample RTM Linking System Requirement to Design Component

Project Name:		Author:	
Date of Review:		Reviewed By:	

Req. ID	Requirement Reference	Requirement Description	Design Reference	System Feature Module Name
APP 1.1	APP SRS Ver 2.1	Better GUI	APP Ver 1.2	Module A
APP 1.2	APP SRS Ver 2.1	Send Alert messages	APP Ver 1.2	Module B
APP 1.3	APP SRS Ver 2.1	Query handling	APP Ver 1.2	Module C
APP 1.4	APP SRS Ver 2.1	Geospatial Analysis	APP Ver 1.2	Module D

Table 2. Sample RTM Linking a Test Case Designed to Test a System Requirement

Unit Test Case #	System Test Case #	Acceptance Test Case #	Requirement Type
APP_GUI.xls	TC_APP_GUI.xls	UAT_APP_GUI.xls	New
APP_MSG.xls	TC_APP_MSG.xls	UAT_APP_MSG.xls	Change Request
APP_QRY.xls	TC_APP_QRY.xls	UAT_APP_QRY.xls	New
APP_GA.xls	TC_APP_GA.xls	UAT_APP_GA.xls	Change Request

approach provides an opportunity for system designers to confirm that the design will meet system requirements. Therefore, in assessing a system design, the design team's approach needs to be examined to see how the team is seeking confirmation of its design in meeting system requirements.

Best Practices and Lessons Learned

Traceability and Verification

Development of project/program scope. The overall goals or desired impact for a project/program must be understood and delineated from the beginning of the effort. The solutions and technologies required can and should evolve during the systems engineering process, but the desired capability end state should be well understood at the beginning. "What problem are we trying to solve?" must be answered first.

Quality of written requirements. Poorly written requirements make traceability difficult because the real meaning is often lost. Ambiguous terminology (e.g., "may," "will") is one way requirements can be difficult to scope, decompose, and test. Assist with written requirements by ensuring a common and clear understanding of terminology.

Excessive reliance on derived requirements. As work moves away from a focus on original requirements, there is a danger of getting off course for performance. Over-reliance on derived requirements can lead to a loss of context and a dilution of the true nature of the need. This is an area where traceability and the bidirectional flow of requirements are critical.

Unique challenges of performance-based acquisition. Performance-based projects present a unique set of issues. The nature of performance-based activity creates ambiguity in the requirements by design. This can be extremely difficult to overcome in arriving at a final, user-satisfactory solution. As a matter of practice, much greater scrutiny should be used in this environment than in traditional project/program development.

Requirements baseline. A requirements baseline is essential to traceability. There must be a trail from the original requirements set to the final implemented and deployed capability. All of the changes and adjustments that have been approved must be incorporated in order to provide a seamless understanding of the end state of the effort. It should also include requirements that were not able to be met. In order to adequately judge performance, the requirements must be properly adjusted and documented.

Project/program risk impacts. The influence of requirements on project/program risk must be evaluated carefully. If sufficient risk is generated by a requirement, then an effective mitigation strategy must be developed and implemented. Eliminating a requirement can be an outcome of this analysis, but it must be weighed carefully. This is an area where an FFRDC trusted agent status is especially critical. Chasing an attractive yet unattainable requirement is a common element in project/program delays, cost overruns, and failures. See the SEG's Risk Management topic.

Watch for requirements that are difficult to test. If requirements are difficult or impossible to test, the requirements can't be traced to results if the results can't be measured. System-of-systems engineering efforts can greatly exacerbate this problem, creating an almost insurmountable verification challenge. The language and context of requirements must be weighed carefully and judged as to testability; this is especially true in a system-of-systems context. See the article "Test and Evaluation of Systems of Systems."

Requirements creep. Requirements creep—both up and down the spectrum—is an enduring conundrum. As requirements flow through the systems engineering process, they can be diluted to make achieving goals easier, or they can be "gold plated" (by any stakeholder) to provide more than is scoped in the effort. Increasing capability beyond the defined requirements set may seem like a good thing; however, it can lead to difficulty in justifying program elements, performance, and cost. Adding out-of-scope capability can drastically change the sustainment resources needed to maintain the system through its life cycle. Requirements creep is insidious and extremely detrimental. On the other hand, the evolution of needs and requirements must be accommodated so that flexibility in creating capabilities can match changing operations and missions and provide timely solutions.

Interaction with end users. Interaction with end users is critical to the requirements traceability and verification cycle. The ability to get feedback from people who will actively use the project or program deliverables can provide early insight into potential performance issues.

Bidirectional requirements traceability. There must be a two-way trace of requirements from the requirements themselves to both larger organizational goals and to applicable capability solutions.

Verification of test plans. Pay careful attention to the development of the requirements verification test plans. An overly ambitious test plan can portray a system that completely meets its requirements as lackluster and perhaps even unsafe. On the other hand, a "quick and dirty" test plan can miss potentially catastrophic flaws in a system or capability that could later lead to personnel injury or mission failure.

Design Assessment

Importance of documentation and team commitment. A thorough review of documentation and an evaluation of the design team's commitment to engage with the stakeholders in the design process are key to conducting a meaningful assessment of whether the system design meets the system requirements.

- Review system development team's strategy/approach to assess team's commitment in meeting system requirements.
  - Interview design team lead and key personnel.
  - Review system documentation.
- Focus assessment review on:
  - Existence of an RTM and its accuracy and currency (this does not have to be exhaustive, but a systematic audit of key system functionality will suffice)
  - Participation in critical design reviews
  - Design team's approach toward outreach to system concept designers and user community (stakeholders)
  - Design team's procedures to capture stakeholder comments
  - Design team's methodology to vet system requirements and process change request.

Importance of documented and validated findings. Document your assessment and validate your findings.

- Re-validate the audit trail of how you arrived at each finding and make any corrections (if needed).
- If possible, consult with design team representative and share key findings.
- Document your re-validated findings and make recommendations.

References and Resources

Agouridas, V., H. Winand, A. McKay, and A. de Pennington, 2008, "Early Alignment of Design Requirements with Stakeholder Needs," *Proceedings of the Institute of Mechanical Engineers*, vol. 222, pp. 1529–1549.

Arkley, P., S. Riddle, and T. Brookes, 2006, "Tailoring Traceability Information to Business Needs," *Proceedings of the 14th IEEE International Conference on Requirements Engineering,* Sept. 11–15.

Chairman of the Joint Chiefs of Staff, March 1, 2009, Joint Capabilities Integration and Development System (CJCSI 3170.01G).

Defense Acquisition Guidebook.

Federal Aviation Administration, May 14, 2007, Systems Engineering Manual.

International Council on Systems Engineering (INCOSE), January 2010, INCOSE Systems Engineering Handbook Version 3.2, INCOSE-TP-2003-002-03.2.

NASA Systems Engineering Handbook.

Ramesh, B., and M. Jarke, January 2001, "Towards Reference Models for Requirements Traceability," *IEEE Transactions on Software Engineering,* vol. 27, issue 1.

Ramesh, B., C. Stubbs, T. Powers, and M. Edwards, April 1995, "Lessons Learned from Implementing Requirements Traceability," *CrossTalk*.

Tran, E., Spring 1999, "Requirements and Specifications," Carnegie Mellon University.

Case Studies

Government Accountability Office, March 2006, Business Systems Modernization, IRS Needs to Complete Recent Efforts to Develop Policies and Procedures to Guide Requirements Development and Management, GAO-06-310.

Minardo, K., September 24, 2007, Tier 1 Case Study of TBMCS/TBONE (1999 - 2006), Social Contexts of Enterprise Systems Engineering MITRE-Sponsored Research Project MITRE Technical Report 070074.

Nordmann, J. C., January 2007, Overview of the Single Integrated Air Picture (SIAP) Enterprise and Lessons Learned to Date (2000 - 2007), An Enterprise Systems Engineering (ESE) Perspective.

Toolkits

FAA Acquisition System Toolset, http://fast.faa.gov, accessed March 12, 2010.

Nordmann, J. C., January 2007, The SEPO Requirements Process Toolkit, accessed March 12, 2010.

Systems Integration

Definition: *Systems integration is the composition of a capability by assembling elements in a way that allows them to work together to achieve an intended purpose.*

Keywords: *acquisition, capability, people, process, program, reward, solution, SoS, system of systems, systems integration*

Context

Systems integration creates a mission capability by composing sub-components of the capability. It is the logical next step between design and development, and testing, verification, validation, and deployment. Always important, integration is increasingly critical to success as the programs MITRE supports migrate to service-oriented, composable-capability architectures.

After components are developed, they must be integrated together with or in the environment in which they are expected to operate. The assembled system is then tested to verify that it performs in accordance with its requirements.

There are different forms of integration. Vertical integration is when the components of a system, developed by a single acquisition program, are integrated to produce the desired capability. Horizontal integration creates new capabilities across individual systems developed by different acquisition programs. Often, the individual systems were originally developed for different customers and purposes. One of the first government resources available on engineering these types of systems is the Systems Engineering Guide for Systems of Systems [1].

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to identify integration and interoperability challenges and create integration strategies that meet the business/mission needs of end-users and their stakeholders. MITRE SEs develop and evaluate integration and interoperability options and observe and assess integration testing. The SE is expected to address integration and interoperability issues associated with the system, including technical, programmatic, social, and business dimensions.

Articles in This Topic

Key aspects of systems integration include (1) identifying and assessing integration and interoperability (I&I) challenges, (2) developing and evaluating I&I solutions, (3) assessing integration testing approaches, and (4) interface management.

The article "Identify and Assess Integration and Interoperability (I&I) Challenges" describes the dimensions of integration and interoperability, the systems engineer's role in addressing them, and best practices and lessons learned in recognizing and evaluating their challenges.

Once the I&I challenges are understood, the SE is expected to develop and evaluate strategies to address them, and recommend a way forward. The article "Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies" discusses the second half of the thread from I&I challenges to solutions. Read the articles together to understand the entire challenge-to-solution thread.

Integration testing often poses significant challenges. The article "Assess Integration Testing Approaches" discusses the problems associated with testing approaches, suggests possible testing strategies and when integration testing, system-of-system testing, or other approaches are appropriate for a program.

Managing interfaces also presents special challenges to systems integration. The article "Interface Management" discusses general principles and best practices of managing interfaces, with special attention to doing so in a complex, interconnected, service-based enterprise, where the consequences of interface decisions can have significant ripple effects.

For complex integration environments, see the article "Systems Engineering Strategies for Uncertainty and Complexity" in the SEG's Enterprise Engineering section.

References and Resources

1. Director, Systems and Software Engineering, Deputy Under Secretary of Defense (Acquisition and Technology), August 2008, Systems Engineering Guide for System of Systems, Ver. 1.0, Office of the Under Secretary of Defense, Acquisition, Technology and Logistics (AT&L), Department of Defense (DoD), Washington, DC.

Additional References and Resources

CMMI Product Team, November 2007, "CMMI® for Acquisition," Version 1.2.

Dahmann, J. S., J. Lane, and G. Rebovich, Jr., November 2008, "Systems Engineering for Capabilities," *CrossTalk—The Journal of Defense Software Engineerin*g.

Hybertson, D. W., 2009, *Model-Oriented Systems Engineering Science*, New York: Taylor & Francis.

Improving processes for acquiring better products and services, Technical Report CMU/SEI-2007-TR-017, ESC-TR-2007-017, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA.

Kelley, M., and R. Kepner, August 2008, Achieving Effective Results in Acquisition – Guidelines for Recommended Action, The MITRE Corporation.

White, B. E., May 12–15, 2008, "Complex Adaptive Systems Engineering (CASE)," *8th Understanding Complex Systems Symposium,* University of Illinois at Champaign-Urbana, IL.

Definitions: *Integration is merging or combining two or more components or configuration items into a higher level system element, and ensuring that the logical and physical interfaces are satisfied and the integrated system satisfies its intended purpose [1]. Interoperability is the ability of two or more systems or components to exchange information and use the information that has been exchanged [2].*

Keywords: *integration, interoperability*

SYSTEMS INTEGRATION

Identify and Assess Integration and Interoperability (I&I) Challenges

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to identify and assess integration and interoperability (I&I) issues in the program they support and in the enterprises of which their system is a part. They are expected to take a broad view of I&I, including technical, organizational, operational, and environmental issues and interactions across systems.

Integration and Interoperability

Identification and assessment of I&I challenges are often system-dependent. Experience shows that integration and interoperability are two sides of the same coin, and SEs need to be concerned about both. Integration is typically addressed when a system is being developed—ensuring that the interfaces are well understood and documented, and that the physical environment has been thoroughly addressed in the design and implementation. Interoperation is more about the role of the developed system—how the various components interact to meet the operational business needs of the customer. A critical first step in identifying and assessing I&I challenges is to understand the systems engineer's responsibilities in addressing integration and the complexities of the associated problems.

Systems Engineering Responsibilities

SEs should adopt the point of view that they own the I&I issues associated with their system, whether they are formally responsible for them or not. This includes all aspects of I&I: technical, programmatic, social, and business. The degree to which any one of these aspects dominates depends on the system being developed, the cultures and agendas of the stakeholder organizations, and the environments into which the system is expected to be deployed.

I&I can cover a broad range of issues, such as:

- Electronic components being incorporated onto a motherboard
- Computer subsystems and software forming a personal computer
- Mechanical components being included in a vehicular drive train
- Radio transceiver and antenna being installed in a vehicle or aircraft
- Multiple computers and applications being built into a command center
- Business interactions crossing many different commands or operating center boundaries.

Common to all of these issues is the need to manage requirements for performance, size, weight, power, and cooling. Environmental constraints must also be considered. When software is a component, processing platforms, operating systems, and languages are all concerns from multiple perspectives, including intended use, future supportability, and modernization roadmaps.

I&I can take on a nested nature. In a worst-case scenario, the SE may have to worry about everything from the development of integrated circuits to the collection of business processes spanning organizations, and everything in between. Fortunately, this extreme situation is rare, and SEs often rely on modern information technology (IT) to provide significant components for their systems. Due to the evolution of IT and the systems engineer's relationship to it, along with the government's increasing focus on capability development, MITRE's role in I&I

is moving from integration of components and subsystems to integration of systems, systems-of-systems, and enterprises.

Complexities of I&I

SEs can identify and assess I&I challenges in several interacting areas: technical, programmatic, social, and business.

The technical area focuses on the system itself, usually without regard for people. There are physical, logical, and environmental aspects to consider. The SE must ensure that physical subsystems fit together and interact properly. Items such as cabling or mechanical fit and finish at the interfaces must be verified. Logically the SE needs to ensure signals are interpreted correctly, and that data exchanged at the interfaces conforms to a defined structure and intended semantics. Further, the SE must understand how the system under consideration fits operationally into the enterprise in which it will exist, and how the technical capabilities work in conjunction with other systems to meet mission needs. Finally, the SE may need to address how the system supports installation and the possibility of dual side-by-side operations with an existing system to support transition.

The programmatic and social areas are dominated by the system stakeholders. Stakeholders include everyone who needs to be involved in the development of the system: owner, administrator, operator, etc. Each will have a different perspective on risk associated with the project, and often these risks are not technical. The SE needs to listen to and consider all stakeholder views, while understanding that the goal is not to accommodate all stakeholder requests. This can be a driver in complexity of system development. Although juggling expectations, budgets, and schedules is a program manager's responsibility, the SE will have a major stake in working out informed decisions.

The SE also must understand the business environment in which the program operates—funding, relationships and dependencies with other programs and organizations, business strategies, and motivations—so integration issues can be identified and understood in the context of this environment.

Finally, enterprise constraints must be addressed. The enterprise is typically concerned with broad-based interoperability, and it levies requirements on developers to help ease integration as the business environment evolves. These restrictions are usually expressed as technical standards to be employed in system development. There are also enterprise processes that will affect the development and fielding of a system (e.g., the Department of Defense Information Assurance Certification and Accreditation Process). It is incumbent on the SE to maintain awareness of enterprise standards and processes. For more information, see the article "Standards Boards and Bodies" in the SEG's Enterprise Engineering section.

SEs are also encouraged to read "Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies," the companion article to this one in the SEG's Systems Integration topic.

Best Practices and Lessons Learned

Interfaces—live or die by them. Interfaces are where the SE can exert control, particularly in the technical area. Internal and external interfaces must be established and their configuration managed. Identify in detail as many interfaces as possible. External interfaces represent the boundary and scope of the system for which the SE is responsible. Interfaces among the stakeholders are equally important. Interfaces to the business process or operations also should not be forgotten.

Communication—essential for success. In addition to providing high-quality documentation and interface specifications that communicate how the system is intended to operate, the SE must also monitor dialog interfaces among the various stakeholders to ensure the program stays on track. Encouragement of open, factual communications among the stakeholders can be the lubrication that makes it all happen. This nontechnical skill is often overlooked.

Subsystems—use them wholesale if possible, but verify their appropriateness. Use of commercial off-the-shelf assemblies for both hardware and software systems is common. To ensure success, all subsystems should be qualified for performance and acceptance-tested commensurate with the risks posed by the component.

Problems in the technical area—plan for the unexpected. Realize and accept that system integration will not be flawless the first time. The more "moving parts" the system has, the bigger the challenge. Provide time in the schedule and funds in the budget to accommodate the occasional failure. Increased testing may also help minimize errors. Close attention to the deployment environment is also warranted. For more information, see the article "Systems Engineering Strategies for Uncertainty and Complexity" in the SEG's Enterprise Engineering section.

Let risk drive your focus. Use risk identification and management strategies to help you decide the specific areas and techniques you will use to focus your work. For example, if the component or subsystem integration activities appear to be well-managed by the contractor compared to some system or enterprise integration issue, focus your attention where it is needed and balance your tasks based on the severity of the risks. For details on risk identification and management strategies, see the Risk Management topic in the SEG's Acquisition Systems Engineering section.

Change—anticipate it. New things are not always welcomed by the people who are expected to use them. A rollout plan and training will be critical to the acceptance of a system. The more stakeholders involved, the greater the degree of difficulty will be.

Unintended system usage—count on it. To accommodate the reality that a system will not be employed exactly as the original designers intended, build in as much flexibility as you can. This is especially true for IT-based systems, where adoption of popular standards at the external interfaces can pay dividends as the system and other systems evolve in their operational environment.

Recognize and address I&I gaps. This is critical, especially when they appear to be outside your area of responsibility. It is the age-old problem of doubles tennis—both players think it is the other's responsibility. System I&I is such a multifaceted and complex area that there is always a risk that an issue or consideration has slipped through the cracks of the integration team. Each SE should take ownership to ensure that I&I occurs thoroughly and correctly across the team's span of influence, both vertically and horizontally.

Standards—a helpful nuisance. Standards-based interfaces are easy to enforce in theory. In reality, they mature over time, compete across standards organizations, and often do not exist for the specialized interfaces you need. Nevertheless, standards provide a meaningful starting place for interface definition, when they are available.

Summary

Integration is a difficult topic due to its many facets. Technical integration has its challenges, but the real difference between success and failure is in dealing with people. Time and energy spent on constructive stakeholder interactions is a good investment for SEs. Well-defined interfaces help the process.

References and Resources

1. Kossiakoff, A., and W. Sweet, 2003, *Systems Engineering: Principles and Practice,* Hoboken, NJ: John Wiley & Sons.
2. Institute of Electrical and Electronics Engineers (IEEE), 1990, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, New York, NY.

SYSTEMS INTEGRATION

Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies

Definitions: *Integration is merging or combining two or more components or configuration items into a higher level system element, and ensuring that the logical and physical interfaces are satisfied and the integrated system satisfies its intended purpose [1]. Interoperability is the ability of two or more systems or components to exchange information and use the information that has been exchanged [2].*

Keywords: *integration, interfaces, interoperability, system(s)*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to develop and evaluate integration and interoperability (I&I) solution strategies for the program they support and the enterprises of which their system is a part. They are expected to take a broad view of I&I, including technical, organizational, operational, and environmental issues and interactions across systems. In general, the MITRE SE "owns" the overall I&I problem space.

Background

Integration and interoperability co-exist with each other. To be successful at integrating two or more elements requires the elements to be interoperable. The elements must co-exist at both the physical and functional levels to be interoperable, and various communication or interface standards must be adhered to. These can be as simple as a 120-volt AC outlet, or as complex as the Transmission Control Protocol and the Internet Protocol that control the exchange of information on a computer network.

As discussed in the companion article "Identify and Assess Integration and Interoperability (I&I) Challenges," to be successful, I&I must consider technical, programmatic, social, and business dimensions.

I&I of information technology (IT)–intensive systems is increasingly important as the concept "the network is the computer" becomes a reality. Almost all products in service today, from televisions to jet aircraft, are either wholly or partially controlled by computers, and therefore I&I strategies for IT-intensive systems require an understanding of two forms of interoperability:

- **Syntactic Interoperability:** If two or more systems are capable of communicating and exchanging data, they are exhibiting syntactic interoperability. Specified data formats, communication protocols, and the like are fundamental. In general, Extensible Markup Language or Structured Query Language standards provide syntactic interoperability. Syntactic interoperability is required for any attempts at further interoperability.
- **Semantic Interoperability:** Beyond the ability of two or more computer systems to exchange information, semantic interoperability is the ability to automatically interpret the information exchanged meaningfully and accurately to produce useful results as defined by the end users of both systems. To achieve semantic interoperability, both sides must agree to a common information exchange reference model, whether it is one used only by themselves or, preferably, a standard model that has been agreed on by a community, so additional interoperability with other systems in the community can be facilitated. Regardless, the content of the information exchange requests is unambiguously defined: what is sent is the same as what is understood.

So what does this mean for the SE in developing and evaluating I&I solution strategies? On the most basic level, it requires an understanding of what is needed to effectively integrate the elements of interest while providing a solution that will be interoperable. The detailed answer to this question should be based on the level of interoperability desired, which needs to be rooted in an assessment of operational needs. A key question is whether syntactic interoperability, semantic interoperability, or both are required. This is not necessarily a simple question to answer, but once it has been, the integration strategy can be developed.

I&I solution strategies must be scrutinized to ensure they satisfy program cost and schedule constraints. Other considerations, such as contract structure, implications to other systems (both the systems that directly interface to the system[s] impacted by the strategies and those that are indirectly impacted), probability of success, and risks must be factored into the evaluation. (Several articles cover subjects relevant to strategy formulation under the Acquisition Program Planning and Program Acquisition Strategy Formulation topics in the SEG's Acquisition Systems Engineering section.)

I&I solution strategies need to account for stakeholder agendas and objectives. This is often called the social or political dimension, and it includes actions to address potential issues or objections certain organizations or personalities may pose. One strategy is to socialize alternate solutions with supporting rationale for preferred options. In doing so, be sure especially to encourage or tease out stakeholder inputs in areas important to them as a way to facilitate their buy-in.

Business considerations should focus on plans and strategies of those stakeholders with the greatest equity in the system. Implications for future work programs, systems, and roadmaps should be part of the evaluation process. Naturally, the operational users should figure prominently in this dimension, but don't forget the government's prime commercial contractors' interests or equities in the system. Factoring them into your business considerations can help shape the I&I strategy so it gives the government better leverage with the contractor.

Best Practices and Lessons Learned

Operational needs are key. The single most important consideration in assessing integration and/or interoperability solutions is to first understand the operational requirement for exchanging data. Ask yourself: "How much and what type of information is required to be exchanged for the system to perform its mission?"

Importance of early and continuous operator involvement. Get users involved early and include them in the operational implications of the I&I solution strategies being considered. This is a continuous process, not a one-time activity.

Working groups that work. Establish an interoperability working group, including all stakeholders, early in the development cycle and meet regularly. There is no better way to ensure nothing "falls through the cracks" than to regularly meet with all stakeholders and discuss the specifics of integration and interoperability. Something as simple as a difference in nomenclature can result in opposite polarities on either end of a signal, with the result that interoperability doesn't happen.

Think broadly. Be sure you step back and consider the broad implications of candidate I&I solutions. Most systems no longer stand alone but are part of one or more systems-of-systems or enterprises. This makes developing a successful I&I solution strategy more difficult. The problems of integrating boards and boxes into a system are being joined and sometimes supplanted by integrating systems into enterprises with only the loosest of defined interfaces. This is of particular importance when an enterprise exists today and a much more sophisticated (i.e., complex) enterprise will replace it, gradually, over time. One example is the FAA/DoD Next Generation Air Transportation System (NextGen) [3] being developed to address the extreme growth in air traffic expected over the next two decades. Much of the responsibility, currently allocated to ground-based systems and air traffic controllers, will move to an airborne network and place more responsibility on cockpit personnel. I&I issues and their potential solutions need to be carefully addressed. These issues involve human cognition and decision-making dimensions, not just questions of enabling technologies.

Think incremental. Consider iterative or incremental steps to achieve the desired level of I&I.

Prototypes and experiments. Consider the use of prototyping and experimentation to understand the I&I environment, especially if the objectives are new or the solution is innovative in nature.

Bottom-up. Work integration issues from the lowest level upward. Do not assume that the "box will work" just because it did in another application, unless all integration aspects have been verified first. Make sure the integration strategy and subsequent verification are documented and agreed on by all involved. Don't arrive at the final test and hear "Well, that's not really what we meant."

References and Resources

1. Kossiakoff, A., and W. Sweet, 2003, *Systems Engineering: Principles and Practice,* Hoboken, NJ: John Wiley & Sons.
2. Institute of Electrical and Electronics Engineers (IEEE), 1990, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, New York, NY.
3. Next Generation Air Transportation System (NextGen) website, http://www.faa.gov/next-gen/, accessed May 24, 2010.

Definition: *When components of a system are developed in isolation, all the pieces must be brought together to ensure that the integrated system functions as intended in its operational configuration. Integration testing should exercise key interfaces between system components to ensure that they have been designed and implemented correctly. In addition, the total operational architecture, including all the segments of the system that are already fielded, should be included in an end-to-end test to verify system integration success.*

Keywords: *end-to-end testing, integration testing, operational architecture, system-of-systems testing*

SYSTEMS INTEGRATION

Assess Integration Testing Approaches

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of integration (or system-of-systems) testing in the acquisition process, where it occurs in systems development, and the benefits and risks of employing it. MITRE SEs are also expected to understand and recommend when integration testing, or system-of-systems testing, is appropriate within a program development. They should be able to take a broader look at the system acquisition within the context of its intended operational environment, beyond simply the core piece of equipment or software that is being developed, to the overarching operational architecture. MITRE SEs should develop and recommend integration testing strategies and processes that encourage and facilitate active participation of end users and other stakeholders in the end-to-end testing process. They are expected to monitor and evaluate contractor integration testing and the acquisition program's overall testing processes, and recommend changes when warranted.

Background

From a software development perspective, system integration testing (SIT) is defined as the activities involved with verifying the proper execution of software components and proper interfacing between components within the solution. The objective of SIT is to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution. While functional testing is focused on testing all business rules and transformations and ensuring that each "black box" functions as it should, SIT is principally focused on testing all automated aspects of the solution and integration touch points [1].

Modern systems provide great value through multifunctionality. However, for the systems engineer, the multifunctionality brings the challenge of increased complexity. Humans deal with complexity by partitioning the challenge into smaller pieces—sometimes called components or modules, although at times these are full systems in and of themselves. The downside of partitioning the problem into manageable pieces is that the pieces have to be put together (integration) and shown to work together. This integration is best achieved through a disciplined systems engineering approach containing good architecture, interface definitions, and configuration management.

In most cases, systems being acquired through the government's acquisition process are not complete, stand-alone entities. The newly acquired system will almost always need to fit into a larger operational architecture of existing systems and/or operate with systems that are being separately acquired. To be completely effective and suitable for operational use, the newly acquired system must interface correctly with the other systems that are a part of the final operational architecture. Integration testing, or system-of-systems testing, verifies that the building blocks of a system will effectively interact, and the system as a whole will effectively and suitably accomplish its mission. This article expands the strict software-focused definition of system integration testing to a broader look at complete systems and the integration, or system-of-systems, testing that should be conducted to verify the system has been "assembled" correctly.

The conundrum that a MITRE systems engineer, or any independent party charged with assessing a system's integration test strategy, will encounter in attempting to recommend or develop integration test strategies is the lack of requirements written at a system-of-systems or operational architecture level. By way of example, although the Department of Defense (DoD) Joint Capabilities Integration and Development System (JCIDS) was developed to address shortfalls in the DoD requirements generation system, including "not considering new programs in the context of other programs" [2], operational requirements documents continue to be developed without a system-of-systems focus. A typical Capabilities Development Document will provide requirements for a system, including key performance parameters, but will not provide requirements at the overarching architecture level. As a result, to develop a recommendation for integration testing, some creativity and a great deal of pulling information from diverse sources are required. Once the test is developed, the task of advocating and justifying the test's need within the system development process will be the challenge at hand.

The following discussion provides examples of systems-of-systems, the recommended integration testing that should be conducted, and both good and bad integration testing examples. Note that best practices and lessons learned are generally interspersed throughout the article. A few cautionary remarks are also listed at the end.

Systems-of-Systems: Definition and Examples

While the individual systems constituting a system-of-systems can be very different and operate independently, their interactions typically deliver important operational properties. In addition, the dependencies among the various systems are typically critically important to effective mission accomplishment. The interactions and dependencies must be recognized, analyzed, and understood [3]. Then the system-of-systems test strategy can be developed to ensure that the integration of the individual systems has been accomplished successfully to deliver a fully effective and suitable operational capability.

The examples used in this article are drawn from a particular domain. But most MITRE SEs should see a great deal of similarity in the essentials of the following examples, regardless of the sponsor or customer they support.

The Global Positioning System (GPS) is an example of a system-of-systems. Typical GPS users—ranging from a hiker or driver using a GPS receiver to navigate through the woods or the local streets, to a USAF pilot using GPS to guide a munition to its target—don't usually consider all the components within the system-of-systems required to guide them with GPS navigation. The constellation of GPS satellites is only a small piece, albeit an important one, within the system-of-systems required to deliver position, navigation, and timing information to the GPS user. Other essential pieces include the ground command and control network needed to maintain the satellite's proper orbit; the mission processing function needed to process the raw collected data into usable information for the end user; the external communication networks needed to disseminate the information to the end user; and the user equipment needed for the end user to interface with the system and use its information. The dependencies and interfaces among all these elements are just as critical to accomplishing the user's goal as is the proper functioning of the constellation of GPS satellites.

A second system-of-systems example is an interoperable and information assurance (IA) protected cross-boundary information sharing environment where federal government users from different departments and agencies, commercial contractors, allies, and coalition members can share information on a global network. Multiple separate but interrelated products comprise the first increment suite of information technology services, including Enterprise Collaboration, Content Discovery and Delivery, User Access (Portal), and a Service-Oriented Architecture Foundation to include Enterprise Service Management.

Finally, an example of a more loosely coupled system-of-systems (SoS)—i.e., a surveillance system-of-systems for which a single government program office is not responsible for acquiring and sustaining the entire SoS. The surveillance network comprises a number of sensors that contribute information in the form of observations to a central processing center (CPC) that uses the sensor-provided observations to maintain a database containing the location of all objects being monitored. The CPC is updated and maintained by one organization while each type of surveillance network contributing sensor has its own heritage and acquisition/sustainment tail. A new sensor type for the surveillance network is currently being acquired. While it will be critically important for this new sensor type to integrate seamlessly into and provide data integrity within the overall surveillance network, the road to SoS integration testing is fraught with difficulty primarily because there are no overarching requirements at the surveillance network level to insure adequate integration of the new sensor.

System–of–Systems Testing

Although they are challenging to plan and execute, system-of-systems tests for programs where a single government program office is responsible for the entire SoS are generally accomplished better as part of the system acquisition process. If nothing else, a final system integration test is typically planned and executed by the development contractor prior to turning the system over for operational testing. Then the operational test community plans, executes, and reports on an operationally realistic end-to-end system test as a part of the system's Congressionally mandated Title 10 Operational Test and Evaluation.

A good example of an integration/SoS test is that being done to inform some GPS upgrades. As the new capability is fielded within the GPS constellation, the development community will combine their Integrated System Test (IST) with the operational test community's Operational Test and Evaluation into an integrated test that will demonstrate the end-to-end capability of the system. This SoS/end-to-end test will include the full operational process, from user request for information, through command generation and upload to the constellation, to user receipt of the information through the user's GPS receiver. During the final phase of the IST, a number of operational vignettes will be conducted to collect data on the end-to-end system performance across a gamut of operational scenarios.

Best Practices and Lessons Learned

- Integration testing should include scenarios that demonstrate the capability to perform mission-essential tasks across the SoS segments.
- Don't assume integration testing will necessarily happen or be adequate just because the full SoS is under the control of a single program office. There are examples of such single program office SoS acquisitions comprising a number of different products and segments that only tested each product separately.
- Failure to conduct adequate SoS integration testing can lead to potentially catastrophic failures. If the new sensor type in the surveillance network example provides the quality and quantity of data anticipated, there is the real possibility that it will overwhelm the CPC's processing capability, thus degrading the accuracy and timeliness of the surveillance database.

Summary and Conclusions

A strict software-development view of integration testing defines it as a logical extension of unit testing [4]. In integration testing's simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process are tested together. Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between them. This method reduces the number of possibilities to a far simpler level of analysis.

This article has focused on making the logical extension of this definition to a full-up system, and expanding the integration testing definition to one of system-of-systems testing. MITRE SEs charged with assessing integration testing approaches should ensure a system-of-systems view of the program, and develop and advocate for full end-to-end testing of capabilities within the complete operational architecture.

References and Resources

1. MIKE 2.0, System Integration Testing, accessed March 5, 2010.
2. Joint Capabilities Integration Development System, Wikipedia, accessed March 5, 2010.
3. System of Systems, Wikipedia, accessed March 5, 2010.
4. DISA/JITC, October 2008, Net-Centric Enterprise Services Initial Operational Test and Evaluation Plan, p. i.

Additional References and Resources

MSDN, Integration Testing, accessed March 5, 2010.

SMC/GPSW, September 9, 2009, Integrated System Test (IST) 2-4 Test Plan draft, pp. 2-21–2-22.

Wikipedia contributors, “United States Space Surveillance Network,” Wikipedia, accessed May 25, 2010.

SYSTEMS INTEGRATION

Interface Management

Definition: *Interface management includes the activities of defining, controlling, and communicating the information needed to enable unrelated objects (including systems, services, equipment, software, and data) to co-function. Most new systems or services require external interfaces with other systems or services. All of these interfaces must be defined and controlled in a way that enables efficient use and change management of these systems or services. Therefore, the practice of interface management begins at design and continues through operations and maintenance.*

Keywords: *change management, coupling, interface*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the general principles and best practices of managing interfaces for Information and Communications Technology (ICT) systems. They are expected to identify the most efficient and effective processes and methods for implementing them, and to understand the complexities that result in an increasingly interoperable world.

Background

Interfaces are the functional and physical connections at the boundaries of ICT systems that are designed to interoperate with other systems. There are many types of interfaces, including communications interfaces, signaling interfaces, service interfaces, data interfaces, hardware interfaces, software interfaces, application program interfaces, etc. These interfaces are critical elements supporting the complex nature of today's systems, which are becoming more geographically distributed and interconnected with other independently developed systems. The strong dependencies on external interfaces require that special attention be given to how they are designed, managed, and publicized beyond what programs typically control within a traditional configuration management process.

The practice of interface management (IxM) is related to requirements and configuration management, but is applied more specifically to the management of interfaces as a subcomponent of ICT systems. IxM is a technical systems engineering activity, focused on the architecture, design, and implementation of the interface. As such, the lead or chief SE typically has primary responsibility for this life-cycle management process [1]. The major inputs and outputs might include:

- **Inputs:** interface management plan, interface requirements, and interface requirements changes
- **Outputs:** interface specifications, interface control documents/drawings, and interface action control sheets.

For this article, the purpose of interface management activity is to:

- Provide all necessary design information needed to enable co-functionality of items, such that separately designed and produced items will be able to work together.
- Use the interface performance, functional, and physical attributes as constraints to design changes, ensuring sustainable use of that interface by an increasing and changing body of users.

Interface Management in an Enterprise Engineering Service–Oriented Environment

Interoperability requires a minimum of two items, each with its own interfaces configured in a way that enables effective joining. A plug and an outlet are examples of that interface. In ICT, interfaces need to be characterized along multiple dimensions, including function, performance, behavioral process, and data. Multiple public standards exist to help with this characterization (such as WSDL, SOAP, UDDI, XML, and KML), but these alone are insufficient. It is necessary to understand how the interface will behave and function, how it can transmit or consume data, and what sequences are required for the exchange of data. Interface management addresses this complexity through use of an engineering management process that is well defined in various engineering bodies of knowledge, such as Software Engineering Body of Knowledge, Software Engineering Institute, International Council on Systems Engineering, and the Defense Acquisition Guidebook.

In a service-oriented environment, the key to successful system or service usage lies in the ability to effectively publicize the requirements to users in a way they understand and need. In addition, users will want some level of assurance or understanding of how that service provider plans to mature and evolve the interface so that users have some level of configuration control over their side of the interface. From a provider's perspective, it may become difficult to understand who is using the service, and the provider may not recognize the scale of the larger set of users that will be affected by any interface changes.

Best Practices and Lessons Learned

The following rules of practice can be used for both internal and external interfaces, as part of the interface management process.

Don't underestimate the need for governance. Interfaces play a crucial role in all systems, which, by definition, consist of multiple components that must interact to deliver a collective capability. Complex systems consist of numerous interfaces of various types; loosely coupled architectures entail higher degrees of abstraction than tightly coupled architectures. In the absence of proper governance, interface sprawl and variation can quickly devolve into degraded system performance, maintainability, and sustainability. Therefore, IxM, which is always important, is considerably more challenging in systems characterized by loosely coupled architectures. The need for comprehensive governance throughout the interface life cycle is essential, and early deliberate planning is necessary.

The major IxM activities needing governance include build-time and run-time contexts. Requirements management, architecture and design standards, technical reviews, and testing are some of the major build-time governance areas, and tend to come from a system development life-cycle perspective. Release management, configuration management, and change management are focus areas for run-time governance, and tend to come from an information technology service management perspective. While it is customary to distinguish between build-time and run-time governance activities, it is clear that life-cycle management spans the two, as change management tends to fold back onto requirements management, and testing unfolds into release management. How an enterprise elects to organize and categorize the governance processes is not as important as the need to ensure that such oversight is executed. Also, recognize that good governance structures mature and change based on the needs of the activities being governed. Deliberately plan a mechanism for periodically reviewing the governance construct for opportunities to improve.

Establish an interface characterization framework and IxM processes early. Often, creation of the interface characterization framework does not occur until after interfaces are developed. As a result, end users are not engaged, the framework is lagging based on information that needs to now be captured as opposed to created during the design phase, and there are other competing priorities for the systems engineer's time (such as delivering the system or service). This results in interfaces that are not well documented, controlled, or publicized. The best time to start is during design, when everyone is thinking about what the system or service should or will be. Create the framework and begin capturing the planned interface information when it is being decided. Establish the teams that are responsible for executing IxM processes to give them time to determine the most efficient way to engage in the systems engineering life cycle. Allocating sufficient time to planning improves the opportunity to deliver good results.

Implement the simplest IxM approach possible. The simplest and most straightforward approach that will satisfy the objective(s) should always be chosen. Complex interface management processes or tools should only be employed when other methods do not meet the needs of those trying to use or manage the interfaces. Why create unnecessary overhead? Engineering resources are scarce and should be leveraged in areas where they are most needed.

Always adhere to standards. Interoperability depends on the use of standards to successfully promulgate and grow a service-oriented environment. The use of standards-based interfaces helps minimize the amount of specialty development and therefore improves the likelihood of service reuse.

Establish service-level agreements for interfaces. Trust is earned. Users need to understand what performance to expect from the interface and its underlying system. Establish service-level agreements (SLAs) to document the performance parameters for the interface and underlying systems, then report performance against those measures. Remember that SLAs are as much about the provider of the interface as they are about the user. If the service level will vary based on the volume of use, then clear articulation of that expected performance variation will help users understand up front, before performance begins to degrade. It will also trigger various management activities to manage that degradation in a thoughtful and planned way.

Use a well-defined framework to describe the interfaces. Having different sets of information attributes for different interfaces complicates and raises the cost of achieving interoperability. It is critical that each system or service categorize and describe the interfaces under their control in a consistent way. In addition, this framework must focus on communicating at different layers of abstraction about how other systems/services need to use the interface. There is resistance to providing additional information beyond that required by some common repository, such as a Universal Description, Discovery, and Integration registry. It is critical that the framework address attributes of using the interfaces (e.g., business process or data flow, SLA, functionality, data map) and not just technical aspects of the interface itself. ITIL suggests using the service catalog as a type of framework to describe services and their interfaces. An interface specification is another alternative. Regardless of which mechanism or framework is used, it is important to include information on current and future versions, their deprecation dates, the technical specification, standards, and other use-related information or tools in one logical place for user access.

Simplify the end-user development challenge with tools where possible. Develop, deploy, and manage development and implementation tools where possible. The best way to support end users is to help them understand when they have correctly configured their end of the interface. For example, conformance test kits can significantly help during the development phase to ensure that inadvertent configuration problems do not arise. This will improve end user satisfaction and increase interoperability and usage. Sample code and even configuration schemas for commercial-off-the-shelf–based interfaces are other tools to consider providing.

Ensure persistent, active engagement of all stakeholders in the IxM process. Users or customers really like to be heard, particularly when they are being asked to make their business dependent on a system or service that is not within their control. So, provide them with that user forum and engage them in IxM activities. Interface Control Working Groups (ICWGs) can be effective mechanisms to publicize and manage interfaces with customer/user participation. Where multiple end users depend on your external interfaces, an ICWG may encourage active and constant participation, which will in turn promote consistency, prevent development of unnecessary interfaces, and reduce risk associated with interface changes. An ICWG is a specialized integrated product team comprising cognizant technical representatives from the interfacing activities. Its sole purpose is to solve interface issues that surface and cannot be resolved through simple engineer-to-engineer interaction.

Plan to deprecate prior versions. It is important not to strand users of prior versions of the interface; however, there is a limit to which backward compatibility should extend. Backward compatibility requirements can constrain future flexibility and innovation and create maintenance and management headaches. However, it is critical for end users to have some level of stability in the interface so they can manage their own development life cycles and mission objectives. Create an opportunity to discuss an interface deprecation strategy and establish a core set of business rules that drive that plan. Ensure that cost/price considerations are understood from both perspectives, provider and end user. Then, publish the business rules in the service catalog or other mechanism in a widely visible and accessible location, alongside other user-related information. This will enable users to manage their side of the interface in a way that does not strand them and also supports growth and innovation.

Publish interface information in an easily accessible and visible location. Often, users do not have ready and easy access to the information they need to use a system or service. Accessibility is required not only for the government, but also their supporting contractors who may or may not have access to particular networks or knowledge centers. Interface information must be readily available from a known location accessible to all resources responsible for making decisions about and developing to the interface. Too often, system and service developers forget that users have different competency levels, and they misunderstand the depth of information users need to be able to effectively configure their systems to interface properly. Good interface management includes understanding the various types of use for the interface information, and presenting the data in a way that supports getting that information easily to the user.

References and Resources

1. Per DoDI 5000.02, Enclosure 12, Section 4.1.6, each Program Executive Officer, or equivalent, shall have a lead or chief systems engineer in charge of reviewing assigned programs' Systems Engineering Plans and overseeing their implementation.

Additional References and Resources

Defense Acquisition University, Defense Acquisition Guidebook, accessed December 10, 2009.

DoD Directive (DODD) 8320.02, Data Sharing in a Net-Centric Department of Defense, April 23, 2007.

DoD Instruction (DoDI) 5000.2, Operation of the Defense Acquisition System, December 8, 2008.

International Council on Systems Engineering (INCOSE), http://www.incose.org/, accessed December 15, 2009.

Military Handbook Configuration Management Guidance MIL-HDBK-61A(SE), February 7, 2001.

National Information Exchange Model, accessed December 10, 2009.

SEPO Configuration Management Toolkit, The MITRE Corporation.

SEPO Enterprise Integration Toolkit, The MITRE Corporation.

Software Engineering Body of Knowledge (SWEBOK), accessed December 15, 2009.

Software Engineering Institute (SEI), Service-Oriented Architectures as an Interoperability Mechanism, July 31, 2009.

Universal Core Overview, https://metadata.ces.mil/ucore/index.html, November 2008.

Test and Evaluation

Definition: *Test and Evaluation (T&E) is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess progress of design, performance, supportability, etc. Developmental test and evaluation (DT&E) is an engineering tool used to reduce risk throughout the acquisition cycle. Operational test and evaluation (OT&E) is the actual or simulated employment, by typical users, of a system under realistic operational conditions [1].*

Keywords: *analysis, DT&E, evaluation, OT&E, performance, testing, verification*

Context

Testing is a mechanism to emsure quality of a product, system, or capability (e.g., right product, built right). To be effective, testing cannot occur only at the end of a development. It must be addressed continuously throughout the entire life cycle.

Test and Evaluation involves evaluating a product from the component level, to stand-alone system, integrated system, and, if appropriate, system-of-systems, and enterprise. Figure 1 highlights these levels of evaluation and how they align with government DT, OT, and accreditation and certification testing.

The articles in this topic are arranged according to a notional V-model life cycle [2] chronology of a program acquisition: (1) creating and assessing T&E strategies, (2) assessing T&E plans and procedures,

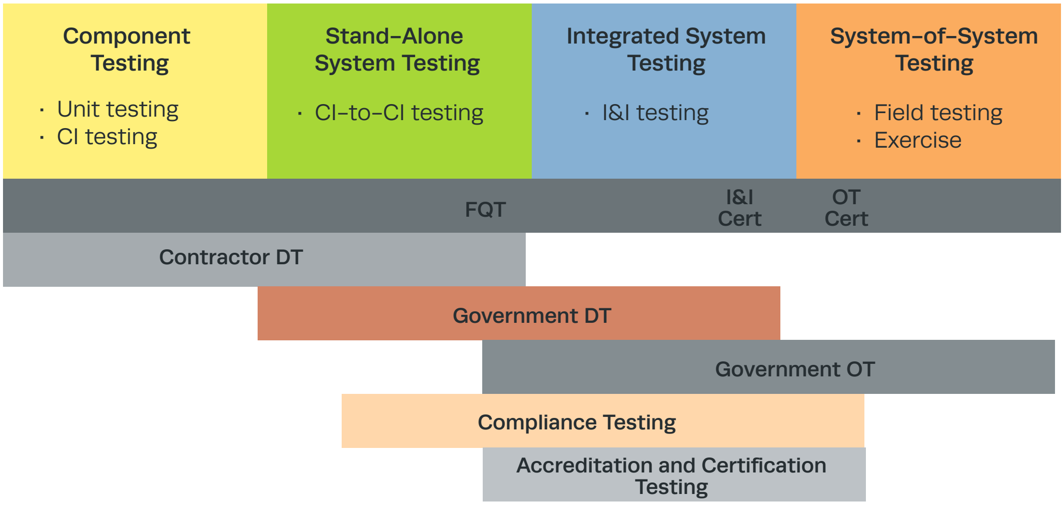


Figure 1. Product Life-cycle Test Phases

(3) verification and validation, and (4) creating and assessing certification and accreditation strategies throughout the process. As noted elsewhere in the SE Guide, the system life cycle is rarely, if ever, as linear as the simplified V-model might imply.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to create test and evaluation strategies to field effective, interoperable systems that include making recommendations on certification and accreditation processes. They assist in developing and defining test and evaluation plans and procedures. MITRE SEs participate in developmental and operational testing, observe and communicate test results, influence re-test decisions, recommend mitigation strategies, and assist the customer/sponsor in making system acceptance decisions.

Best Practices and Lessons Learned

Employ prototypes and M&S to advantage. Prototypes and/or modeling and simulation (M&S) used early in a program can help predict system performance and identify expected results, both good and bad. Both techniques can be used in designing, evaluating, or debugging portions of a system before incurring the expense of "bending metal."

Common sense—sometimes a rarity. Use common sense in testing. For example, although it is necessary to ensure that environment testing covers the environment that your system is expected to operate in, specifying and testing a system to operate to -70°C when it will be used in an office-like environment is a sure way to either fail the test or drive the cost of the system beyond reason. This is an especially common pitfall when designing systems for mobile or airborne environments. Extreme temperature, vibration, radiated emissions, etc., are not always encountered in these environments. Ensure that the tests are realistic.

Match testing method with purpose. There are many forms of testing. Some involve instrumented measurements of system performance during "live" operations. Others, in decreasing order of complexity, are analysis, demonstration, or inspection. Select the testing method that suits the purpose. The performance of a critical operational capability (e.g., identification of airborne objects as friend, hostile, or neutral) will likely require all or most of the methods and culminate in a "live" test. Analysis is suited to testing requirements like long-term reliability of electronic components, and when assessing inspection is appropriate (e.g., number of operator consoles in a command center). Selecting the right verification methods produces the right results and saves time and cost.

Test strategy—start early and refine continually. Plan the test strategy from the onset of the program and refine it throughout the program's life cycle. Involve the right stakeholders in the development and review of the test strategy and plans.

Don't overlook the basics. Ensure that tests have been developed to be objective and capable of assessing compliance with a requirement. Make sure that if one test is intended to validate many lower level requirements, you are sufficiently versed with the details of the system design and have the results of the component level tests available. This is particularly important in preparing for operation testing.

Ready or not? Determining suitability to enter a test is a key decision that can substantially affect the overall success of the program. Know when you are ready, and know when you are not. When unsure, postponing or deferring testing may be the most prudent long-term course of action.

References and Resources

1. Defense Acquisition University website, https://acc.dau.mil/t&e.
2. Wikipedia contributors, "V-Model," Wikipedia, accessed January 13, 2010.

Additional References and Resources

Defense Acquisition University, "Test and Evaluation," *Defense Acquisition Guidebook*, Chapter 9.

Haimes, Y., 1998, *Risk Modeling, Assessment, and Management*, Wiley & Sons.

International Test and Evaluation Association website, http://itea.org/.

Stevens, R., R. Brook, K. Jackson, and S. Arnold, 1998, *Systems Engineering: Coping with Complexity*, Prentice Hall.

The MITRE Institute, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model, Ver. 1," Section 2.6.

Definition: *A Test and Evaluation strategy "...provide[s] information about risk and risk mitigation, ...[and] empirical data to validate models and simulations, evaluate technical performance and system maturity, and determine whether systems are operationally effective, suitable, and survivable... [1]."*

Keywords: *evaluation, governance, strategy, test*

TEST AND EVALUATION

Create and Assess Test and Evaluation Strategies

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) work with sponsors to create or evaluate test and evaluation strategies in support of acquisition programs. They are often asked to recommend test and evaluation approaches, which provide insights that can be used to manage acquisition risks. They also monitor government and contractor test and evaluation processes, and recommend changes when they are warranted. Subsequent to contract award, MITRE SEs evaluate test and evaluation master plans produced by both contractors and government test organizations. They also evaluate test plans and procedures that are applied during development testing, operational testing, and for some customers, live fire testing; occasionally, they help to formulate the plans and procedures as a member or advisor to the government test team. As a consequence, MITRE SEs are expected to understand the rationale behind the requirement for acquisition programs to create and execute a test and evaluation strategy. They are expected to understand where test and evaluation activities such as interoperability testing, information assurance testing, and modeling and simulation fit in the acquisition life cycle, and where they can be used most effectively to identify and mitigate risk. Finally, it is expected that MITRE SEs, in the course of their other activities such as requirements and design analysis, will include test and evaluation concerns in their analysis.

Background

The fundamental purpose of test and evaluation (T&E) is to:

> provide knowledge to assist in managing the risks involved in developing, producing, operating, and sustaining systems and capabilities. T&E measures progress in both system and capability development. T&E provides knowledge of system capabilities and limitations to the acquisition community for use in improving the system performance, and the user community for optimizing system use in operations. T&E expertise must be brought to bear at the beginning of the system life cycle to provide earlier learning about the strengths and weaknesses of the system under development. The goal is early identification of technical, operational, and system deficiencies, so that appropriate and timely corrective actions can be developed prior to fielding the system. [1]

The program manager is responsible for creating and submitting a test and evaluation strategy after the decision is made to pursue a materiel solution. The creation of the test and evaluation strategy involves planning for technology development, including risk; evaluating the system design against mission requirements; and identifying where competitive prototyping and other evaluation techniques fit in the process.

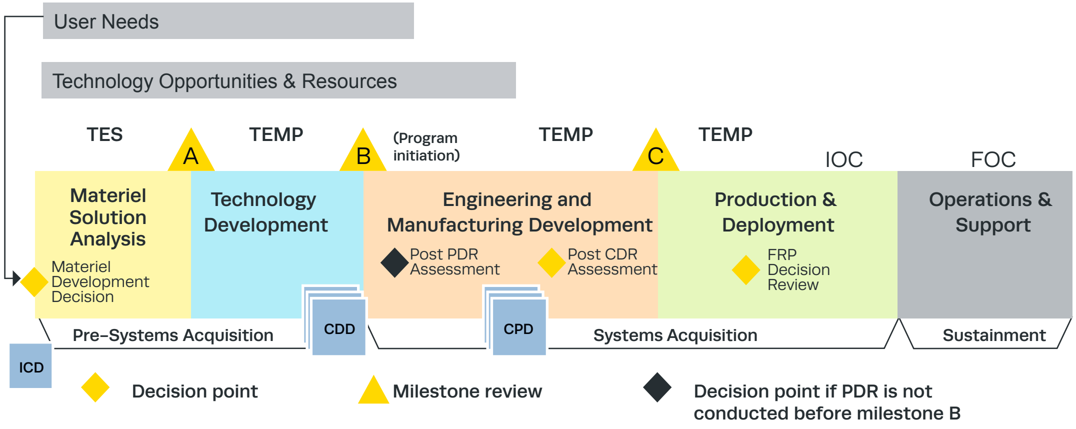
The content of a test and evaluation strategy is a function of where it is applied in the acquisition process, the requirements for the capability to be provided, and the technologies that drive the required capability. A test and evaluation strategy should lead to the knowledge required to manage risks; the empirical data required to validate models and simulations; the evaluation of technical performance and system maturity; and a determination of operational effectiveness, suitability, and survivability. In the end, the goal of the strategy is to identify, manage, and mitigate risk, which requires identifying the strengths and weaknesses of the system or service being provided to meet the end goal of the acquisition program. Ideally, the strategy should drive a process that confirms compliance with the Initial Capabilities Document (ICD), instead of discovering later that functional, performance, or non-functional goals are not being met. The discovery of problems late in the test and evaluation phase can have significant cost impacts as well as substantial operational repercussions.

Historically, test and evaluation consisted of testing a single system, element, or component, and was carried out in a serial manner. One test would be performed, data would be obtained, and then the system would move to the next test event, often at a new location with a different test environment. Similarly, the evaluations themselves were typically performed in a serial manner, with determinations of how well the system met its required capabilities established through the combination of test results obtained from multiple sites with differing environments. The process was time-consuming and inefficient, and with the advent of network-centric data-sharing strategies, it became insufficient. In large part this was due to an approach to acquisition that did not easily accommodate the incremental addition of capabilities. Creating and maintaining an effective test and evaluation strategy under those conditions would have been difficult at best. A test and evaluation strategy is a necessity today because of the addition of capabilities via incremental upgrades, which is now the norm, and the shift to a network-centric construct where data is separated from the applications; data is posted and made available before it is processed; collaboration is employed to make data understandable; and a rich set of network nodes and paths provide the required supporting infrastructure.

When there is a need to deliver a set of capabilities as quickly as possible, further complexity in creating a test and evaluation strategy can be introduced, especially in cases where ICDs are largely nonexistent, ambiguous, inconsistent, or incomplete. In this situation, the development of a test and evaluation strategy represents a significant challenge, and in some cases it may be largely ignored to get a capability in the field as quickly as possible. However, this approach is not without attendant risk assessments and mitigation strategies—they are just accomplished at a high level very early in the process. Quick reaction capabilities (QRCs) of this sort are often followed by a more formal acquisition effort, a program of record. Nonetheless, test and evaluation of QRCs cannot be completely ignored. At the outset, the critical capabilities must be identified, and their risks must be identified, managed, and mitigated through some level of test and evaluation.

Government Interest and Use

Government acquisition communities are recognizing the need for a test and evaluation strategy that is in concert with evolving department and agency network-centric data-sharing strategies. Although a test and evaluation strategy is created early in the acquisition process (Figure 1), it has to be refined as the acquisition process evolves and system details become more specific. A test and evaluation strategy needs to be developed early in the acquisition process to ensure that it is consistent with the acquisition strategy, identifies the required



Technology development strategy (TDS) Test evaluation strategy (TES) ID emerging T&E capability requirements ID T&E requirements in RFP Annual report	Test and evaluation master plan (TEMP) Execute T&E program Provide T&E results for OIPT/DAB CDD requirements for testability and evaluation TRL evaluation T&E requirements in RFP Annual report	Test and evaluation master plan (TEMP) Execute T&E program Provide T&E results for OIPT/DAB Support SE tech review (PDR) CPD requirements for testability and evaluation Define system capabilities and limitations Discovery and deficiencies Annual report	Test and evaluation master plan Verification of corrections for deficiencies T&E results for OIPT/DAB OTRR IOT&E Annual report	Follow-on DT and OT Verification of corrections for deficiencies Develop T&E programs to support upgrades, modifications, increments

Figure 1. T&E in the Defense Acquisition Management System [4]

resources (facilities, ranges, personnel, and equipment, including government-furnished equipment), encourages shared data access, engages the appropriate government test agencies, identifies where and when modeling and simulation will be employed, and establishes both the contractor's and government's test and evaluation efforts.

MITRE can and should influence how a test and evaluation strategy evolves and is applied, and, in particular, should ensure that it is consistent with the acquisition strategy and the systems engineering plan, if there is one. It is rare for MITRE, or any other single organization, to be asked to independently create a test and evaluation strategy. It is far more common for MITRE to collaborate with the government stakeholders to create a test and evaluation strategy, or to be employed to evaluate and recommend changes to a strategy that is the product of a test and evaluation working group or other test and evaluation stakeholder organization. In these instances, it is important that MITRE become a collaborator and consensus builder.

In most instances, the government establishes a working group to execute the test and evaluation strategy. This group is often referred to as a test and evaluation working integrated product team, and it consists of test and evaluation subject matter experts from the program office, customer headquarters, customer user representatives, test and evaluation organizations, higher oversight organizations (e.g., Office of the Secretary of Defense for DoD systems), supporting FFRDCs, and other stakeholders. The test and evaluation strategy is a living document, and this group is responsible for any updates that are required over time. The program manager looks to this group to ensure that test and evaluation processes are consistent with the acquisition strategy and that the user's capability-based operational requirements are met at each milestone in the program. Finally, as a program progresses from pre-systems acquisition to systems acquisition, the test and evaluation strategy begins to be replaced by a test and evaluation master plan, which becomes the guiding test and evaluation document (Figure 1). The DoD's interest in and application of a test and evaluation strategy is documented in Incorporating Test and Evaluation into Department of Defense Acquisition Contracts [2] and Chapter 9 of the Defense Acquisition Guidebook [3].

Best Practices and Lessons Learned

New thinking required for T&E in net-centric and SOA environments. The transition to network-centric capabilities has introduced new test and evaluation challenges. Network capabilities can reside in both nodes and links, and the basic system capabilities can reside in service-oriented architecture (SOA) infrastructures, with the remaining capabilities provided by services that are hosted on the SOA infrastructure. The test and evaluation of capabilities in this type of framework requires new thinking and a new strategy. For example, evaluating the performance of the network itself is probably not going to be accomplished without extensive use of modeling and simulation because the expense of adding live nodes in a lab increases dramatically with the number of nodes added to the test apparatus. This places a greater burden on the veracity of the modeling and simulation because one of the keys to obtaining the metrics that will support risk mitigation is gaining an understanding of the effect of a new host platform on the network infrastructure, as well as the effect of the network infrastructure on the new host platform. A test and evaluation strategy that mitigates risk in the development of a network infrastructure that will support network-centric warfare requires a balance of theoretical analysis and laboratory testing. MITRE can help develop a strategy that employs a mix of modeling and simulation that has been verified, validated, and accredited; laboratory testing; and distributed testing that takes advantage of other network-enabled test components and networks. The capabilities required to execute a network-centric test and evaluation strategy have evolved over the past few years, and today we have a rich set of networks (such as the DREN the and SDREN) that host nodes that constitute government laboratories, university facilities, test centers, operational exercise sites, contractor facilities, and coalition partner facilities.

There are emerging technology aspects of the network-centric transformation where test organizations have limited experience, and these aspects are where MITRE can help create and assess test and evaluation strategies. These new technology areas constitute the heart of the SOA that will make up the enterprise, as well as the services themselves that make up new capabilities.

Accounting for governance in T&E. The transition to a service-based enterprise introduces some new complexities that must be accounted for in the test and evaluation strategy. Service-based enterprises rely on a more formalized business model for the identification of required capabilities. While this is not a new concept, the formalization of business processes into the engineering process, and the addition of the concomitant governance, add new complexities to both the systems engineering and test and evaluation processes. A test and evaluation strategy must account for governance of capabilities (e.g., services) as well as the capabilities themselves. Service repositories become critical parts of the test and evaluation strategy and must encompass how services are distributed, populated, managed, and accessed, since a critical aspect of service-based capabilities is reuse of existing services to compose new capabilities.

Accounting for business process re-engineering and scalability of service-based infrastructure in T&E. The shift to network-centric service-based enterprise capabilities is rarely accomplished in a single stroke; instead it is accomplished incrementally, beginning with business process re-engineering and the identification of scalable service-based infrastructure. Both of these activities need to be incorporated into the test and evaluation strategy, and their evaluation should begin as early as possible. Prototyping or competitive prototyping are common techniques used to evaluate service-based infrastructures, especially the ability of the infrastructure to scale to meet future needs and extend to accommodate future capabilities.

The importance of factoring in refactoring. Business process re-engineering leads to segregating capabilities into those that will be provided by newly developed services, and those that will be provided by refactored legacy components. It also enables a block and spiral upgrade strategy for introducing new capabilities. An evaluation of how it is decided which capabilities will be newly developed and which will be refactored legacy components is critical to the health of the program and should constitute another early and critical aspect of the test and evaluation strategy. Each legacy component selected for refactoring must be analyzed to determine how tightly coupled it is to both the data and other processes. Failure to do so can lead to the sort of "sticker shock" some current programs have experienced when attempting to add capabilities through spiral upgrades.

Distributed test environments. A key distinction of, and enabling concept in, the network-centric service-based construct is the ability to reuse capabilities through a process referred to as finding and binding. Achieving the true acquisition benefits of service-based programs requires that capabilities that can be reused be discoverable and accessible. To do this, service registries must be established and a distributed test environment be employed, which in turn places new requirements on the test and evaluation strategy for these types of programs. Distributed test and evaluation capabilities must be planned for, resourced, and staffed, and shared data repositories must be established that will support distributed test and evaluation. Network infrastructures exist that host a wide variety of nodes that can support distributed test and evaluation (e.g., DREN and SDREN). However, early planning is required to ensure they will be funded and available to meet program test and evaluation needs.

Importance of metrics for loose coupling in T&E strategy. Another area where a test and evaluation strategy can be effective early in a service-based acquisition program is in the continuous evaluation and measurement of the loose coupling that maintains separation of data and applications, and enables changes in services with minimal impact to other services. The average contractor business model leans toward tight coupling simply because it ensures that the contractor is continuously engaged throughout the program's life cycle. Failure to establish and apply metrics for loose coupling as part of the test and evaluation strategy will lead to a lack of insight into system performance; the impact of tight coupling with respect to interfaces will be unknown until the interfaces are actually in play, which is often too late to mitigate the risk involved. Consequently the test and evaluation strategy must include an identification and metrics-based analysis of interfaces to mitigate the risk that data and applications are tightly coupled; the earlier this is accomplished, the easier it is to mitigate the problem.

Data sharing implications for T&E strategy. Often overlooked in development and test and evaluation of service-based enterprises are the core capabilities for data sharing. While time is devoted to the test and evaluation of services that enable data sharing, the underlying technologies that support it are often not brought into the test and evaluation process until late. The technologies critical to data discovery and sharing are embedded in metadata catalog frameworks and ontology products, both of which require a skill set that is more esoteric than most. The consequence of this is that aspects of discovery and federation through the use of harmonized metadata are overlooked, and instead individual contractor metadata is employed for discovery. This leads to a downstream need for resource adapters that bridge metadata used in one part of the enterprise or for one type of data to other parts of the enterprise. In several instances, the downstream requirement for resource adapters has ballooned to account for nearly every data store in the enterprise. A test and evaluation strategy that incorporated the harmonization of metadata, the development of a single ontology, and the early test and evaluation of these items would have saved time and money, and delivered a capability to the warfighter earlier.

Summary

The shift to a network-centric data-sharing strategy has introduced a new set of challenges in the acquisition process. Incremental development of capabilities has become the norm, and distributed enterprise capabilities are the desired end-state. Test and evaluation must evolve to keep pace with the shift in development processes. In this article we have captured a few of the best practices and lessons learned, but the list could go on at length to include those practices that still provide significant risk identification, management, and mitigation. In addition, as information technology in particular evolves, the risk areas will shift and coalesce, driving the need for new and updated test and evaluation strategies.

References and Resources

1. Department of Defense Instruction Number 5000.02, December 8, 2008, Operation of the Defense Acquisition System, USD(AT&L), Enclosure 6 Integrated T&E, p. 50.
2. *Incorporating Test and Evaluation into Department of Defense Acquisition Contracts*, 2009, Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Washington, DC.
3. *Defense Acquisition Guidebook*.
4. DoD Presentation: Test and Evaluation Working Integrated Product Team, August 17, 2009.

Additional References and Resources

Department of Defense Directive Number 5000.01, May 12, 2003, The Defense Acquisition System, USD(AT&L).

Definition: *Test and evaluation is the set of practices and processes used to determine if the product under examination meets the design, if the design correctly reflects the functional requirements, and if the product performance satisfies the usability needs of personnel in the field.*

Keywords: *acceptance test, integration test, operational test, peer reviews, system test*

TEST AND EVALUATION

Assess Test and Evaluation Plans and Procedures

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be familiar with different kinds of tests, which group conducts the tests, how to evaluate test documents, and the developer's procedures for test control. MITRE SEs are also expected to analyze test data.

Background

Testing is the way a product, system, or capability under development is evaluated for correctness and robustness, and is proved to meet the stated requirements. Testing is done at each stage of development, and has characteristics unique to the level of test being performed. At a macro level, testing can be divided into developer testing conducted before the system undergoes configuration management, and testing conducted after the system undergoes configuration management. Testing done before configuration management includes peer reviews (sometimes called human testing) and unit tests. Testing done after configuration management includes integration test, system test, acceptance test, and operational test. An operational test is normally conducted by government testing agencies. The other tests are conducted by the developer; in some cases, such as acceptance test, government observers are present.

Assessing Test and Evaluation Plans and Procedures

Assessment normally begins with the Test and Evaluation Master Plan (TEMP), which is the driver for much of what follows. The TEMP is developed by the government; detailed test plans and procedures are created by the developer. The scope, direction, and content of the TEMP are driven by the nature of the program, the life cycle, the user needs, and the user mission. For example, testing software developed for the program is quite different from testing systems that are largely based on, and require considerable integration of, commercial-off-the-shelf (COTS) products. The TEMP will influence the testing documents produced by the developer, but the developer's documents are largely driven by what it produces and is working to deliver.

The government program management office (PMO) is tasked with assessing the developer's test and evaluation plans and procedures. Often MITRE plays a central role in helping the PMO perform this assessment. The requirements on which the developer's test plans and procedures are based must be well crafted. A valid requirement is one that is measurable and testable. If it is not measurable and testable, it is a poor requirement. Developer test plans and procedures should be based on the functional requirements, not the software design. Both the test community within the developer organization and the development community should base their products on the functional requirements.

When assessing the developer's test plans and procedures, the focus should be the purpose of the test—that is, to assess the correctness and robustness of the product, system, or service. The tests should prove the product can do what it is intended to and, second, can withstand anomalous conditions that may arise. This second point requires particular care because there are huge differences in how robustness is validated in a COTS-based system versus software developed for a real-time embedded system. The environment in many

COTS-based business systems can be tightly bound. A name or address field can be limited in terms of acceptable characters and field length. In a real-time embedded system, you know what the software expects to receive if all is going as it should, but you do not always know what possible input data might actually arrive, which can vary in terms of data type, data rate, and so on. Denial-of-service attacks often try to overwhelm a system with data, and the developer's skill in building robustness into the system that allows it to handle data it is not intended to process has a great deal to do with the eventual reliability and availability of the delivered product. It is not unusual for the error protection logic in complex government systems to be as large as, or larger than, the operational software.

Assessment of the test plans and procedures must take all of these issues into account. The assessor must understand the nature and purpose of the system and the kind of software involved, and must have the experience to examine the test plans and procedures to assure they do an appropriate job of verifying that the software functions as intended. The assessor must also verify that, when faced with anomalous data conditions, the software will respond and deal with the situation without crashing. The test conditions in the test plans and procedures should present a wide variety of data conditions and record the responses.

For software systems, especially real-time systems, it is impossible to test all possible paths through the software, but it should be possible to test all independent paths to ensure all segments of the software are exercised by the tests. There are software tools to facilitate this, such as the McCabe suite that will identify paths as well as the test conditions needed to put into a test case. However it is accomplished, this level of rigor is necessary to assure the requisite reliability has been built into the software.

Unlike the unit test, the integration test plans and procedures focus on the interfaces between program elements. These tests must verify that the data being passed between program elements will allow the elements to function as intended, while also assuring that anomalous data conditions are dealt with at their entry point and not passed to other programs within the system. The assessor must pay particular attention to this when assessing the integration test plans and procedures. These tests must be driven by the functional requirements, because those drive what the software must do for the system to be accepted by the sponsor.

Test and Evaluation Phases

Pre–Configuration Management Testing

The two primary test practices conducted prior to configuration management are:

- **Peer Reviews:** Peer reviews are performed to find as many errors as possible in the software before the product enters the integration test. They are one of the key performance activities at Level 3 of the Software Engineering Institute's (SEI) Capability Maturity Model. The SEI accepts two kinds of peer reviews: code walkthroughs, and software inspections (the SEI preferred process, sometimes called Fagan Inspections in reference to Mike Fagan, who developed the process). Software inspections have a well-defined process understood throughout the industry. Done properly, software inspections can remove as much as 87 percent of the life-cycle errors in the software. There is no standard process for walkthroughs, which can have widely differing levels of rigor and effectiveness, and at best will remove about 60 percent of the errors in software.

- **Unit Test:** The developer conducts the unit test, typically on the individual modules under development. Unit test often requires the use of drivers and stubs because other modules, which are the source of input data or receive the output of the module being tested, are not ready for test.

Post-Configuration Management Testing

Testing conducted after the product is placed under developer configuration control includes all testing beyond unit test. Once the system is under configuration management, a problem discovered during testing is recorded as a trouble report. This testing phase becomes progressively more expensive because it involves integrating more and more modules and functional units as they become available; the system therefore becomes increasingly more complex. Each test requires a documented test plan and procedure, and each problem encountered is recorded on a trouble report. Each proposed fix must be validated against the test procedure during which it was discovered, and must also verify that the code inserted to correct the problem does not cause another problem elsewhere. With each change made to respond to a problem, the associated documentation must be upgraded, the fix must be documented as part of the configuration management process, and the fix must be included in the next system build so that testing is not conducted with patches. The longer it takes to find a problem, the more rework is likely, and the more impact the fix may have on other system modules; therefore, the expense can continue to increase. Thus performing good peer reviews and unit tests is very important.

- **Integration Test:** This is a developer test that is successively more complex. It begins by integrating the component parts, which are either the modules that have completed the unit test or COTS products, to form functional elements. The integration test progresses from integration of modules to form entire functional elements, to integration between functional elements, to software-hardware integration testing. Modeling and simulation are often used to provide an operational-like testing environment. An integration test is driven by an integration test plan and a set of integration test procedures. Typically an integration test will have embedded within it a subset of tests identified as regression tests, which are conducted following a system build. Their objective is to verify that the build process did not create a serious problem that would prevent the system from being properly tested. Often regression tests can be automated.

- **Test Data Analysis:** When conducting peer reviews, unit tests, integration testing, and system tests, a significant amount of data is collected and metric analysis is conducted to show the condition state of the system. Significant metric data is produced related to such things as defect density, pass-fail data on test procedures, and error trend analysis. MITRE SEs should be familiar with test metrics, and they should evaluate the test results and determine the likelihood of the system being able to meet the requirements of performance delivered on time and within budget.
- **System Test:** This is an operational-like test of the entire system being developed. Following a successful system test, a determination is made whether the system is ready for acceptance test. After the completed system test, and before the acceptance test, a test readiness review (TRR) may be conducted to assess the readiness of the system to enter the acceptance test.
- **Acceptance Test:** Witnessed by the government, this is the last test before the government formally accepts the system. Similar to the system test, the acceptance test is often a subset of the procedures run during system test.
- **Operational Test:** Performed by an operational unit of the government, this is the final test before the system is declared ready for general distribution to the field.

Best Practices and Lessons Learned

- Examine the reports on the pre-configuration management tests to evaluate the error density information and determine the expected failure rates that should be encountered during subsequent test periods.
- Review the peer review and unit test results prior to the start of integration testing. Due to the expense and time needed to correct problems discovered in the post-configuration management tests, the SE should understand how thorough the prior tests were, and whether there is a hint of any issues that need to be addressed before the integration test starts.
- If peer reviews and unit tests are done properly, the error density trend data during the integration test should show an error density of 0.2 to 1.2 defects per 1,000 source lines of code.
- Consider modeling and simulation options to support or substitute for some aspects of integration that are either of lower risk or extremely expensive or complex to perform with the actual system.
- Complete a thorough independent review of the test results to date prior to sup-

porting the TRR. This is especially true for performance or design areas deemed to be of the greatest risk during the design phase. Once the TRR is passed and the program enters acceptance testing, correcting problems is extremely expensive and time-consuming.

- Involve the government Responsible Test Organization (RTO) early (during the concept development phase is not too early) so they understand the programmatic and technical issues on the program. Including the RTO as part of the team with the acquisition and engineering organizations will lessen conflicts between the acquisition organization and RTO due to lack of communication and misunderstanding of objectives.

References and Resources

"Best Practices," SEPO Library via Onomi, accessed March 15, 2010.

"Capability Maturity Model Integration (CMMI)," SEPO Library via Onomi, accessed March 15, 2010.

"Commercial Off the Shelf Software (COTS)," SEPO Library via Onomi, accessed March 15, 2010.

Department of Justice, Test and Evaluation Master Plan, accessed May 28, 2010.

Federal Aviation Administration, System Process Flowcharts/Test and Evaluation Process and Guidance, accessed May 28, 2010.

"Interoperability," SEPO Library via Onomi, accessed March 15, 2010.

"System Test Plan," SEPO Library via Onomi, accessed March 15, 2010.

Tufts University, Department of Computer Science, *The Test Plan*, accessed May 28, 2010.

Definition: *Verification is "the process for determining whether or not the products of a given phase of development fulfill the requirements established during the previous phase [1]." Validation is the "evaluation of the capability of the delivered system to meet the customer's operational need in the most realistic environment achievable [1]."*

Keywords: *systems engineering life cycle, validation, verification*

TEST AND EVALUATION

Verification and Validation

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand where verification and validation fit into the systems engineering life cycle, and how to accomplish them to develop effective and suitable systems. They are expected to assist in developing and defining requirements, specifications for test and evaluation plans, and verification and validation procedures. MITRE SEs are expected to participate in developmental and operational testing, observe and communicate test results, influence retest decisions, recommend mitigation strategies, and assist the customer/sponsor in making system acceptance decisions. They are expected to evaluate test data and verify that specified requirements are met and validated to confirm operational capabilities.

Background

MITRE's government customers may describe the systems engineering life-cycle model differently. The Department of Defense (DoD) customer uses the DoD 5000.02 process to describe a "five-stage" systems engineering life cycle [2]. This DoD 5000.02 life cycle model maps to other equivalent models described (e.g., International Organization for Standardization [ISO] 15288 Systems and Software Engineering Life Cycle Processes, and Institute of Electrical & Electronics Engineers [IEEE] 1220-2005 Standard for Application and Management of the Systems Engineering Process). Figure 1, as depicted in the "Engineering for Systems Assurance" Manual Ver. 1.0 published by the National Defense Industrial Association (NDIA), shows the interrelationships between the different life-cycle processes [3].

Regardless of the life-cycle model our customer uses, they all track to three basic systems engineering stages: concept development, engineering development, and post-development. Each of these engineering stages may be separated into supporting phases. The concept development phase is critical because it describes the ultimate operational requirements that will be used to "validate" the ultimate material solution. The supporting system, subsystem, and component-level requirements leading to preliminary design and critical design will be iteratively verified through various types of testing and analysis during materialization, integration, and testing. Verification is the critical feedback element that confirms the requirements specified in the previous phase were satisfied. Validation is final confirmation that the user's needs were satisfied in the final material solution. It cannot be overemphasized that Verification and Validation (V&V) and Test and Evaluation (T&E) are not separate stages or phases, but integrated activities within the SE process. Figure 2, from the Washington State Department of Transportation (DOT), illustrates how V&V provide feedback to the systems engineering process [4].

Government Interest and Use

Using the ISO 15288 Systems and Software Engineering Life Cycle Processes as a model, V&V are critical activities that are executed continuously throughout the process. During the initial concept development stage, verification activities confirm that the operational and performance requirements and functional specifications are viable. These requirements and specifications may be developed by the government, by MITRE, and/or by other Systems Engineering and Technical Assistance (SETA) contractors and must be verified. The operational requirements will be used by the government for ultimate validation of the material solution. The performance and functional specification must also be validated because they will be used by the developing contractor to drive preliminary and critical design and to develop the material solution. During the engineering development stage, the subcomponents and components that comprise the material solution must be verified, integrated, and tested.

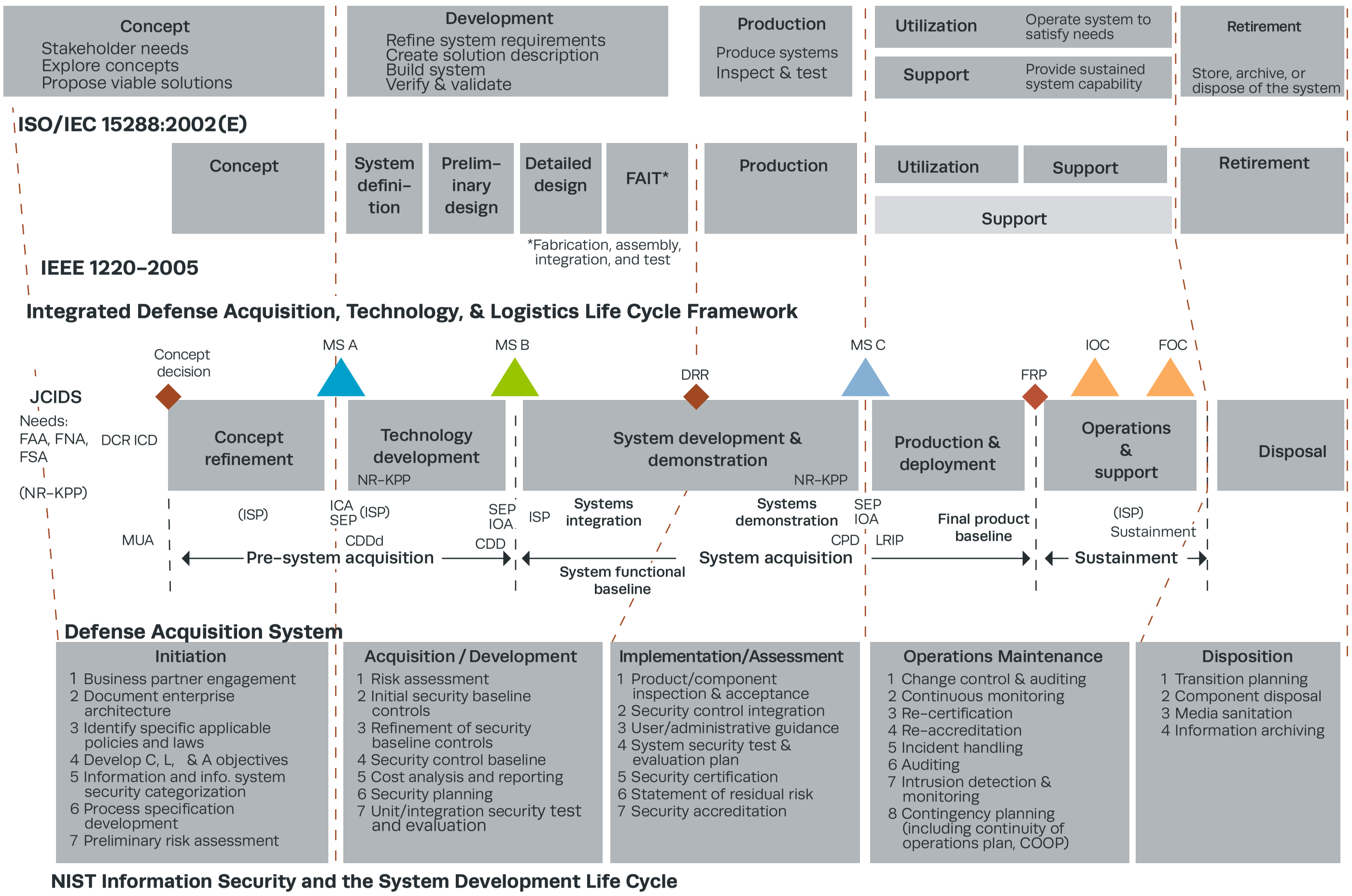

Concept
Stakeholder needs
Explore concepts
Propose viable solutions
Development
Refine system requirements
Create solution description
Build system
Verify & validate
Production
Produce systems
Inspect & test
Utilization
Operate system to satisfy needs
Support
Provide sustained system capability
Retirement
Store, archive, or dispose of the system
ISO/IEC 15288:2002(E)
Concept
System defini–tion
Prelim–inary design
Detailed design
FAIT*
Production
Utilization
Support
Support
Retirement
*Fabrication, assembly, integration, and test
IEEE 1220–2005
Integrated Defense Acquisition, Technology, & Logistics Life Cycle Framework
Concept decision
MS A
MS B
DRR
MS C
FRP
IOC
FOC
JCIDS
Needs:
FAA, FNA, FSA
DCR ICD
(NR–KPP)
Concept refinement
Technology development
NR–KPP
System development & demonstration
NR–KPP
Production & deployment
Operations & support
Disposal
MUA
(ISP)
ICA SEP
(ISP)
CDDd
SEP IOA
CDD
ISP
Systems integration
Systems demonstration
SEP IOA
CPD
LRIP
Final product baseline
(ISP)
Sustainment
Pre–system acquisition
System functional baseline
System acquisition
Sustainment
Defense Acquisition System
Initiation
1 Business partner engagement
2 Document enterprise architecture
3 Identify specific applicable policies and laws
4 Develop C, L, & A objectives
5 Information and info. system security categorization
6 Process specification development
7 Preliminary risk assessment
Acquisition / Development
1 Risk assessment
2 Initial security baseline controls
3 Refinement of security baseline controls
4 Security control baseline
5 Cost analysis and reporting
6 Security planning
7 Unit/integration security test and evaluation
Implementation/Assessment
1 Product/component inspection & acceptance
2 Security control integration
3 User/administrative guidance
4 System security test & evaluation plan
5 Security certification
6 Statement of residual risk
7 Security accreditation
Operations Maintenance
1 Change control & auditing
2 Continuous monitoring
3 Re–certification
4 Re–accreditation
5 Incident handling
6 Auditing
7 Intrusion detection & monitoring
8 Contingency planning (including continuity of operations plan, COOP)
Disposition
1 Transition planning
2 Component disposal
3 Media sanitation
4 Information archiving
NIST Information Security and the System Development Life Cycle

Figure 1. DoD Life–Cycle Framework, and National Institute of Standards and Technology Information Security and the System Development Life Cycle

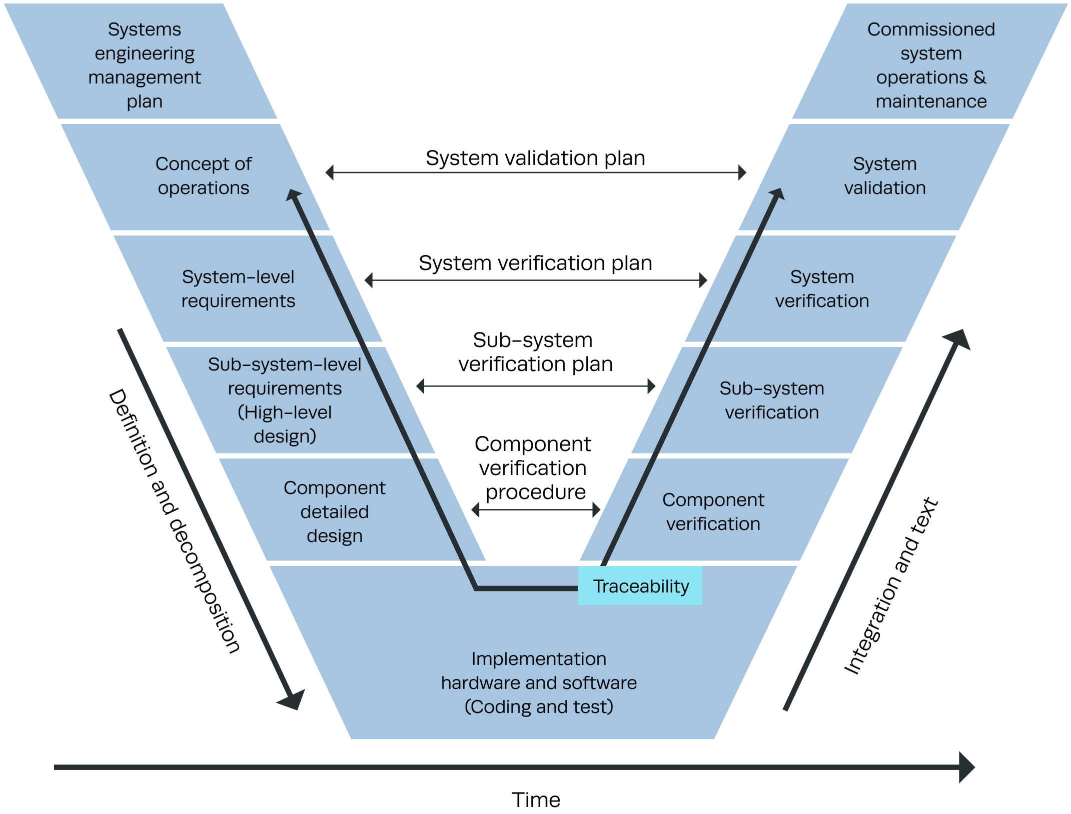

Systems engineering management plan
Concept of operations
System-level requirements
Sub-system-level requirements (High-level design)
Component detailed design
Implementation hardware and software (Coding and test)
Commissioned system operations & maintenance
System validation
System verification
Sub-system verification
Component verification
System validation plan
System verification plan
Sub-system verification plan
Component verification procedure
Traceability
Definition and decomposition
Integration and text
Time

Figure 2. Systems Engineering "V," Washington State DOT

Operational testing is the venue that gathers data to validate that the ultimate material solution satisfies required operational capabilities. V&V are critical activities that confirm that the "contracted for" material solution provides the required operational capability.

Best Practices and Lessons Learned

Continuously coordinate process execution. In many cases, the capabilities development, systems acquisition, and systems engineering processes, although interdependent, are executed independently by different organizations (or different parts of the same organization, such as the prime contractor). Disconnects between the activities executed by the different stakeholders can create serious problems. This can lead to cost and schedule overruns, and reduced operational capability. Active verification of the output from each step of the process and an active risk management program can go far to identify and address disconnects as they occur. The earlier a problem is identified and addressed, the less expensive the resolution will be in terms of cost, schedule, and performance. Early and continuous involvement of subject matter experts is required.

Operational requirements verification—a team sport. Verified operational requirements are critical to validation of the system. In many cases, the operational requirements are poorly documented or change during the course of an acquisition. Verification of operational requirements must involve all potential stakeholders, including the acquisition program manager, systems engineering team, and validation agent (operational tester).

Smart contracting. The government develops operational capability needs, functional requirements, and systems specifications that are placed on contract to develop preliminary and critical designs, and to materialize the system. The contract should include Contract Data Requirements Listings (CDRLs) that require the contractor to develop, and the government to approve, test plans; monitor test execution; and deliver reports that support subcomponent, component, and system verification. This may involve additional upfront cost for the program. However, failure to do so is likely to result in additional cost, longer schedule, and performance shortfalls to the required operational capability. The acquisition program manager, SE, and government contracting officer must work carefully to shape requests for proposal, evaluate responses, and form the contract to include these CDRLs.

Harmonize use of modeling and simulation (M&S) in verification. M&S can be used as part of the T&E process to verify subcomponents, components, and systems. The program manager should involve the contractor, and development and operational test agencies to identify where M&S can be used to generate data for use in verification. Establish the intended use of M&S by each of the testing stakeholders at the beginning of the systems engineering process. The M&S approach can then be harmonized across several intended users and phases of V&V.

Integrated testing supports continuous verification and operational validation. The goal of Operational Test and Evaluation (OT&E) is to confirm that the "concept" developed on the left side of the systems engineering "V" can be validated in the "material solution" on the right side. The Operational Testing Agent (OTA) often seeks contractor and developmental test data to validate capabilities. Often requirements cannot be validated because CDRLs were not specified in the contract and/or developmental test data were not clearly specified by the OTA or delivered by the program manager/developer. In some cases, the verification activities were haphazard or not properly executed. In cases where there has been an undisciplined approach to verification, test and evaluation (missing entry or exit criteria for each step), significant gaps and holes may exist in the material solution that are not evident until OT&E is executed. CDRLs, integration, security, interoperability, development test events, and supporting data requirements should be clearly specified in T&E Master Plans. Time to fix and retest also would be included in the process. The ultimate goal is to execute an integrated testing approach where a component/system/ system-of-systems testing and verification can be executed by one stakeholder, and the data accepted by all stakeholders. Any such testing/verification approach must be documented in test and evaluation plans and resources to execute documented.

References and Resources

1. Kossiakoff, A., and W. Sweet, 2003, *Systems Engineering: Principles and Practice*, John Wiley and Sons, ISBN 0-471-23443-5
2. U.S. Department of Defense, December 8, 2008, "Operation of the Defense Acquisition System," DoD Instruction 5000.02.
3. National Defense Industrial Association System Assurance Committee, "Engineering for System Assurance," Ver. 1.0.
4. Washington State DOT, July 2010, *WSDOT Design Manual*, Chapter 1050.03, Systems Engineering: Systems Engineering "V."

TEST AND EVALUATION

Create and Assess Certification and Accreditation Strategies

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the principles of certification and accreditation (C&A), how a government development organization initiates the C&A process, and how the government sponsor maintains accreditation status following product delivery. They are also expected to understand information assurance (IA) and C&A requirements and processes so they can advise when the government or the contractor is not complying with the letter or intent of department or agency policies and processes. MITRE SEs are expected to understand how systems engineering decisions may impact the IA posture of a system.

Definition: *"Certification is the comprehensive evaluation and validation of a[n] ... information system (IS) to establish the degree to which it complies with assigned information assurance (IA) controls based on standardized procedures. An accreditation decision is a formal statement by a designated accrediting authority (DAA) regarding acceptance of the risk associated with operating a[n] ... IS and [is] expressed as an authorization to operate (ATO), interim ATO (IATO), interim authorization to test (IATT), or denial of ATO (DATO) [1]."*

Keywords: *accreditation, certification, DIACAP*

Introduction

This article is intended to provide general guidance on C&A of all government systems. It follows the Department of Defense (DoD) C&A process and is directly applicable to DoD systems. C&A processes for other U.S. government systems are similar in their essentials but otherwise may vary. In the latter case, the guidance presented here should serve as a general reference for the conduct of C&A activities. Non-DoD department or agency guidance should always take precedence for C&A of their systems.

Certification and Accreditation Process Overview

C&A processes applied to federal and DoD systems are similar. These similarities include use of a common set of functional roles as shown in Table 1.

Table 1. Functional Roles

Role	Function/Responsibility
Information Owner	An official with statutory or operational authority for specified information and responsibility for establishing the controls for its generation, collection, processing, dissemination, and disposal.
Information System Owner	Individual, group, or organization responsible for ensuring the system is deployed and operated according to the agreed-on security requirements.
Certifying Authority/Agent (CA)	Individual, group, or organization responsible for conducting a security certification, or comprehensive assessment of the management, operational, and technical security controls in an information system.
Designated Accrediting Authority (DAA) or Authorizing Official	An official with the authority to formally assume responsibility for operating a system at an acceptable level of risk.

The following generic C&A process overview is based on the functional roles described above.

1. The information owner establishes data sensitivity and security protection requirements.
2. The information system owner implements technical, administrative, and operational security controls in accordance with security protection requirements provided by the information owner.
3. The CA evaluates the security controls incorporated by the system and makes a recommendation to the DAA on whether the system satisfies its security requirements.

4. The DAA assesses the residual security risk, based on the CA's recommendation, and makes an accreditation decision.
5. The information system owner operates the accredited system, which must undergo periodic review and/or re-accreditation.

DoD Information Assurance Certification and Accreditation Process (DIACAP) [2, 3]

DIACAP is the C&A process applied to systems that store or process DoD information. It is defined in DoD Instruction 8510.01 as the "process to manage the implementation of IA capabilities and services and provide visibility of accreditation decisions regarding the operation of DoD information systems (IS), including core enterprise services and Web services–based software systems and applications." [1]

In supporting C&A of a system, MITRE should help the program manager (PM) assemble the DIACAP team, identify requirements, design solutions, implement the system, and integrate testing. The entire DIACAP team should be assembled at program inception to determine the IA Strategy, to agree on the mission assurance category (MAC) and confidentiality level, negotiate a baseline set of IA controls, and assign responsibilities. If there is no team review of system design for compliance with IA requirements, then testing of IA and functional requirements, which sometimes can conflict, will likely not be integrated. It is important that the DIACAP team be assembled to resolve discrepancies throughout the acquisition life cycle; without that cooperation, it is more likely the PM or engineers will make unilateral decisions the DAA may not be able to accept. To help ensure a successful positive C&A outcome, MITRE, often acting as "lead integrator" for the activity, should at the outset reach back to staff members who support the CA and DAA to ensure coordination and agreement regarding the scope of the C&A process.

Process Artifacts

Execution of the DIACAP produces a number of engineering artifacts that are summarized in Table 2.

These artifact documents, together with all other documents resulting from the DIACAP process, are typically produced by the program office and/or the acquisition team. When a contractor produces a DIACAP document, it is reviewed and approved by the program office, often with a MITRE SE involved.

Data Sensitivity and Mission Assurance Category

Each DoD system can be characterized by two pieces of information: the confidentiality level of the data that it processes and its MAC. These characteristics drive the selection of IA controls that the system must implement and the level of robustness (i.e., strength of mechanism) required.

The confidentiality level of a system is based on the highest classification or sensitivity of information stored and processed by the system. The confidentiality level is expressed in three categories: public, sensitive, and classified. More stringent authentication, access control, and auditing requirements apply to systems that process classified data than to systems that process sensitive data, while systems that process public data enforce minimal authentication, access control, or auditing requirements.

Table 2. Engineering Artifacts

Artifact	Description
System Information Profile (SIP)	Information to register about the system being developed.
DIACAP Implementation Plan (DIP)	Enumerates, assigns, and tracks the status of IA controls being implemented.
DIACAP Scorecard	Records the results of test procedures/protocols used to validate implemented IA controls.
Plan of Action & Milestones (POA&M)	Identifies tasks or workarounds to remediate identified vulnerabilities.
Supporting Certification Documents	A compilation of IA controls validation artifacts provided to the CA.
Interim Approval to Test (IATT)	An accreditation decision is a special case for authorizing testing in an operational information environment or with live data for a specified time period.
Interim Approval to Operate (IATO)	An accreditation decision intended to manage IA security weaknesses while allowing system operation for up to 180 days, with consecutive IATOs totaling no more than 360 days.
Denial of Approval to Operate (DATO)	An accreditation decision that the system should not operate because the IA design, IA controls implementation or other security is inadequate and there are no compelling reasons to allow system operation.
Approval to Operate (ATO)	An accreditation decision for a system to process, store, or transmit information for up to three years; indicates a system has adequately implemented all assigned IA controls and residual risk is acceptable.

As described in DoD Instruction 8500.2, p. 22, the MAC assesses the value of the system "relative to the achievement of DoD goals and objectives, particularly the warfighters' combat mission [3]." For systems designated as MAC I systems, loss of integrity or availability would result in immediate and sustained loss of mission effectiveness and cannot be tolerated. MAC II systems also have stringent data integrity requirements but may be unavailable for short periods of time without impacting mission effectiveness. With MAC III systems, the loss of data integrity and/or availability has no significant impact on mission effectiveness or operational readiness.

Information Assurance Controls

Information assurance controls used in DIACAP are detailed in DoD Instruction 8500.2 [3]. These safeguards are grouped into IA baselines, where the selection of an IA baseline is governed by the confidentiality level and MAC of the system. Table 3 specifies the number of IA controls that a system must satisfy as a function of the sensitivity and MAC level of that system.

Table 3. IA Controls Requirements

Sensitivity/MAC	MAC III	MAC II	MAC I
Public	75	81	81
Sensitive	100	106	106
Classified	105	110	110

These numbers reflect the upper bound on the number of IA controls; in reality, many of the IA controls in a given IA baseline may not apply or may be inherited from an external, interconnected system. It should also be noted that while the number of IA controls required for MAC I and MAC II are the same for a given sensitivity level, the level of effort to satisfy those controls is often significantly higher for MAC I systems, where system availability requirements are considerably more stringent.

If IA requirements are not identified early in the acquisition/development process at the time functional requirements are identified, IA requirements cannot be built into the system and tested along with functional requirements. Although inappropriate, C&A is often performed after the system has been built; therefore, when IA controls validation is performed and the C&A documentation is presented to the CA and/or DAA, missing IA requirements may be identified far too late in the acquisition life cycle. It is much more costly to modify a system to comply with IA requirements after it has been built than it is to build in IA up front.

Robustness

A number of IA controls specify system robustness, which DoD Instruction 8500.2 defines as “the strength of mechanism and assurance properties of an IA solution.” IA robustness requirements are expressed by IA controls as basic, medium, or high, and depend on the MAC and sensitivity level of the system and the threat environment in which the system will be deployed.

Commercial off-the-shelf (COTS) IA products or IA-enabled products selected for use in a system must satisfy IA robustness requirements established for that system. The robustness of COTS IA products is evaluated through the National Information Assurance Partnership (NIAP) [4]. An NIAP evaluation assigns an evaluated assurance level (EAL) rating to each product as a means for selecting IA products for use in system acquisition or development programs. Table 4 summarizes the IA characteristics of each robustness level and associated product EAL ranges.

Table 4. IA Characteristics

Characteristics	**Robustness Level**		
	Basic	**Medium**	**High**
General Description	Commercial-grade best practice	High-end commercial-grade	High assurance design
Access Control	Authenticated access control	Strong (e.g., PKI-based) authenticated access control	NSA-endorsed access control and key management capabilities
Key Management	NIST-approved key management	NSA-approved key management	
Cryptography	NIST FIPS-validated cryptography	NIST FIPS-validated cryptography	NSA-certified cryptography
Protection Profiles	Assurance properties consistent with NSA-endorsed basic robustness protection profiles	Assurance properties consistent with NSA-endorsed medium robustness protection profiles	Assurance properties consistent with NSA-endorsed high robustness protection profiles, where available
Evaluated Assurance Level (EAL)	EAL1 – EAL3	EAL4 – EAL5	EAL6 – EAL7

Best Practices and Lessons Learned

IA, C&A, and security in general are not viewed as fundamental requirements and are often traded off when program funding is limited or cut (IA is not funded as a separate line item in the budget). PMs and engineers often don't realize that IA requirements are critical to the functionality of a system; IA ensures the appropriate amount of confidentiality, integrity, and availability are built in—something the warfighter demands.

Developmental testing (DT) is often performed in a vacuum. IA controls are not identified early on; therefore, the IA testing cannot be integrated into the DT. DT is planned and performed without consideration of the Operational Test & Evaluation (OT&E) certification requirement. Deficiencies are identified in OT&E that should have been caught in DT and fixed.

Once an ATO is issued, the tendency is to place the C&A package on the shelf for three years until the next accreditation is needed. If the system is not monitored constantly, with new vulnerabilities mitigated as they are discovered and as threats become increasingly more advanced, the IA posture of the system quickly degrades.

Employ information systems security engineering (ISSE) and reference the IA Technical Framework. The intent is for ISSEs to work with SEs throughout the acquisition life cycle to build IA into the system. This cooperative effort will yield functional capability that is also secure. The SE and ISSE effort will also yield documentation that can be used as evidence of compliance with assigned IA controls—no need to generate special documents just to satisfy the CA and DAA.

Don’t wait until the system is completely built to begin testing the IA controls. As capability is developed, test it—integrate the DT with IA controls testing. Also, integrate the DT with OT&E. OT&E teams can reuse the results of DT, but perform their own analysis. The test and evaluation master plan should identify, integrate, and track all types of testing.

If the CA can’t participate in the DIACAP team meetings, employ the agent of the certifying authority (ACA) (or Service equivalent). The ACAs were established to stand in for the CA and handle the day-to-day certification activities. The ACAs also perform hands-on validation of IA controls, not a desktop review as may be done at a headquarters level. The ACAs are trusted by the CA, and the CA can take a DIACAP scorecard at face value—the CA need not dig into the details of a C&A package, so staffing goes faster.

PMs (with the help of the SE, ISSE, and ACA) must build a realistic POA&M. In conjunction with the DIACAP scorecard, the POA&M should accurately convey the residual risk to the DAA. The PM must aggressively resolve weaknesses and constantly update the POA&M, submitting it quarterly to the DAA for review/acceptance.

Keep the system current, relevant, and secure with a robust incident response and vulnerability management program. Threats evolve and exploit new vulnerabilities, thereby increasing risk. Constantly monitor the system to identify changes (threats, vulnerabilities, operations, environment, etc.) that could impact the IA posture. Ensure the IA manager is a member of the configuration control board to review all changes impacting IA.

References and Resources

1. DoDI 8510.01, November 28, 2007, "DoD Information Assurance Certification and Accreditation Process (DIACAP)."
2. Information Assurance Certification and Accreditation Process (DIACAP).
3. DoD Instruction 8500.2, February 2003, "Information Assurance (IA) Implementation."
4. The NIAP Evaluated Products List.

Implementation, O&M, and Transition

Definitions: *Implementation is the realization of a system (application, plan execution, idea, model, design, specification, standard, algorithm, or policy) into an operational environment.*

O&M (Operations & Maintenance): *When a system is fielded, it enters an operations phase. Preventive maintenance is a schedule of actions aimed at preventing breakdowns and failures before they occur and at preserving and enhancing equipment reliability by replacing worn components before they fail. When a system fails, corrective maintenance is performed.*

Transition is when the system moves from development to the manufacturing/fielding/sustainment phase.

Keywords: *corrective, implementation, maintenance, operations, preventive, transition*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to take into account successful sustainment of their system during pre-acquisition and acquisition phases. They are expected to be able to develop transition strategies for delivering and deploying systems, including simultaneous system operation, cutover, and retirement/disposal of systems to be decommissioned. SEs develop technical requirements and strategies to enable and facilitate system operation, maintenance, and operator training, and evaluate those developed by others. MITRE SEs develop approaches to enable system modifications and technology insertion.

Context

Although most of sustainment is typically not within MITRE's primary systems engineering purview, activities during the concept development, design, and verification phases, where MITRE does have significant influence, can promote or inhibit successful operation and sustainment of fielded systems.

Discussion

Although no underlying articles exist yet for this topic, see the topic-level best practices and lessons learned below. When available, the articles in this topic will describe best practices and lessons learned for transitioning a system from its acquisition phase to the remainder of its life cycle.

Best Practices and Lessons Learned

View transition as a process. Though it is common to think of the transition from development to fielding as a point in time, as described in this article's definition, transition is actually a process that takes place over time. The transition process is a set of activities that encompasses (1) planning for transition, (2) implementation, and (3) O&M in a sequential, phased order. The transition process should not be deferred until after the product is developed and ready to be produced and fielded, or the likelihood of failure is greater. Begin planning the transition process early in the development phase to account for any uniqueness in manufacturing, fielding, or maintenance activities. Deciding how to insert technology improvements in the O&M phase, often necessary multiple times, needs to be addressed in the initial design and development of the system.

Start planning for the transition early in the product development phase, even before the initial design review. Figure 1 represents a Department of Defense perspective of a product's life cycle, and is fairly representative of any product development/manufacture/sustainment processes. As indicated in the figure, the transition strategy needs to begin as early as possible, even during the product planning/strategy sessions. Include manufacturing engineers and field-service engineers in the product planning phase to ensure that the development engineers understand how the product will be produced and maintained. Minor changes in the design, if done early on, can provide a significant benefit to the person who must build or maintain the product.

Walk the manufacturing floor. Have the SEs tour the manufacturing or integration facility and understand how a product "flows" through the plant. Many times, simple modifications to the process can greatly improve the quality of the product being built. Talk to the personnel who

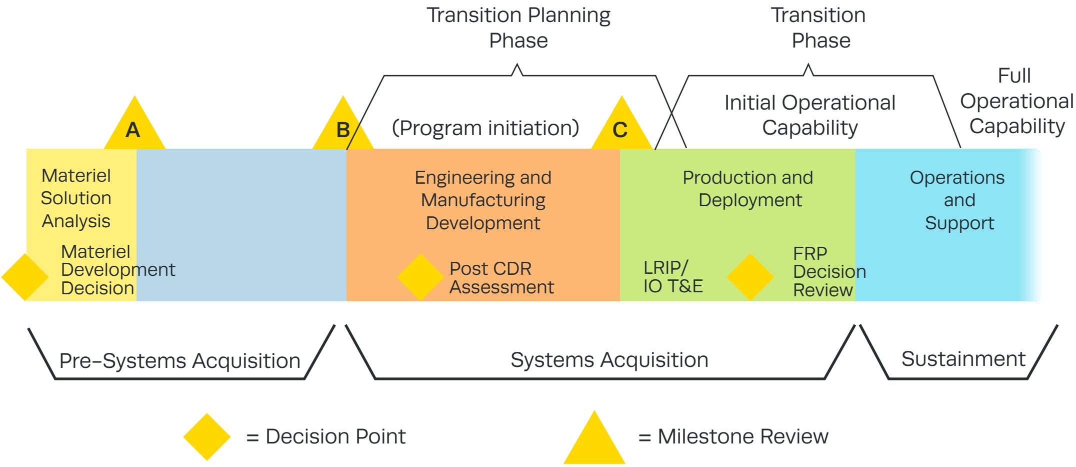

Transition Planning Phase
Transition Phase
Full Operational Capability
Initial Operational Capability
(Program initiation)
A
B
C
Materiel Solution Analysis
Materiel Development Decision
Engineering and Manufacturing Development
Post CDR Assessment
Production and Deployment
LRIP/ IO T&E
FRP Decision Review
Operations and Support
Pre-Systems Acquisition
Systems Acquisition
Sustainment
= Decision Point
= Milestone Review

Figure 1. Transition Activities in Acquisition Life Cycle

work on the floor, to understand what works well and what does not. The more you understand about the actual manufacturing and integration process, the smoother the transition will be.

Understand the customer's maintenance concept. In the commercial world, a field engineering force typically services all aspects of the system. In the government sector, "two-level" maintenance "boxes" are typically swapped in the field (many times by the user), and boxes needing repair are sent back to a repair depot or contractor facility. Regardless of the method used, knowing the responsibilities of maintenance personnel and the tools at their disposal helps develop a product that is easier to maintain.

Have a flexible technology roadmap. Understand and have a plan for when, how, and why technology improvements should be inserted into the system after deployment. Understand technology readiness assessments and develop a roadmap that balances technology maturity, the ability to insert the technology into the system (e.g., from a manufacturing or retrofit perspective), and when it is operationally appropriate to make the technology upgrade.

References and Resources

Defense Acquisition University, Systems Engineering, *Defense Acquisition Guidebook*, Chapter 4.

Defense Acquisition University, Life Cycle Logistics, *Defense Acquisition Guidebook,* Chapter 5.

Other SE Life-Cycle Building Blocks Articles

This topic is a staging area for articles on subjects of relevance to SE Life-Cycle Building Blocks that don't neatly fit under one of its other topics. In most cases, this is because the subject matter is at the edge of our understanding of systems engineering, represents some of the most difficult problems MITRE systems engineers work on, and has not yet formed a sufficient critical mass to constitute a separate topic. This is definitely not the "et cetera" pile of articles.

As more articles are added and an organizing theme among several of them becomes evident, a new topic will be created. In other cases, if a connection between an article and an existing topic becomes apparent, the article will be moved and the topic description revised to reflect the change. Last, a subject may remain a critical, special area of interest and a related article will continue to be carried under this topic.

The article "Spanning the Operational Space—How to Select Use Cases and Mission Threads" covers use cases and mission threads as useful mechanisms to document an existing system's functionality or to establish a user's needs for a new system. They reflect functional requirements in an easy-to-read format. Use cases and mission threads are used during many stages of system development (e.g., to capture requirements, to serve as the basis for the design process itself, to validate the design, for testing, etc.).

The article “Acquiring and Incorporating Post-Fielding Operational Feedback into Future Developments: The Post-Implementation Review” follows the fielding of a system and informs future efforts. As such, it follows the last life-cycle phases that we have defined here.

The article “Test and Evaluation of Systems of Systems” addresses test and evaluation plans in the SoS environment to help MITRE SEs apply systems engineering processes and make plans for systems that are constituents of systems of systems (SoS).

The article “Verification and Validation of Simulation Models” describes best practices and lessons learned for developing and executing a plan to verify and validate a simulation model of a system or system component, and in collecting and analyzing different types of verification and validation data.

The article “Affordability, Efficiency, and Effectiveness (AEE)” describes the various practices and analyses that MITRE SEs can apply to achieve three key success measures—affordability, efficiency, and effectiveness (AEE)—in developing and shaping engineering solutions, making program recommendations, and evaluating engineering efforts on behalf of their sponsor’s mission.

OTHER SE LIFE-CYCLE BUILDING BLOCKS ARTICLES

Spanning the Operational Space—How to Select Use Cases and Mission Threads

Definition: *Use cases describe a system's behavior ("who" can do "what") as it responds to outside requests. The use case technique captures a system's behavioral requirements by detailing scenario-driven threads through functional requirements.*

A mission scenario/thread (an instance type of a use case) represents one path through the use case. In a sequence diagram, we can show one thread for the main flow through the use case and others for possible flow variations through it (e.g., from options, errors, breaches).

Keywords: *behavioral requirements, functional requirements, mission description, mission execution, operational requirements, process description, sequence diagram, software engineering, systems engineering, UML*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and roles of use cases and mission threads, where to employ them in an acquisition program life cycle, and the benefits and risks of using them. MITRE SEs are expected to understand when use cases and mission threads are appropriate to a situation and to develop detailed recommendations and strategies for using them. They are expected to monitor and evaluate contractor activities in using mission threads and use cases, as well as in the government's overall strategy, and recommend changes when warranted.

Introduction

The use case technique captures a system's behavioral requirements by detailing scenario-driven threads through the functional requirements.

A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Actors are parties outside the system that interact with the system. An actor may be a class of users, roles users can play, or other systems. A primary actor is one having a goal requiring the assistance of the system. A secondary actor is one from which the system needs assistance.

Mission threads are associated with one or more operational vignettes. A vignette describes the overall environment: the geography, organizational structure and mission, strategies, tactics, and information on any protagonists, including their composition, strategies, tactics, and timing. Mission threads can describe tactical operations, logistical operations, support operations, and maintenance, training, test, and development operations. Mission threads serve as drivers for developing the architecture and as the basis for test cases during a verification cycle.

What They Are and When to Use Them

Use cases and mission threads are a useful mechanism to document an existing system's functionality or to establish user needs for a new system. Use cases describe the system from the user's point of view and the interaction between one or more actors. (The interaction between the actor[s] and the system is represented as a sequence of simple steps.) Actors may be end users, other systems, or hardware devices. Each use case is a complete series of events described from the point of view of the actor. One or more scenarios or threads may be generated from a use case. The aggregate of all threads documents the detail of each possible way of achieving the system's goal. Use cases typically avoid technical jargon, preferring instead the language of the end user or domain expert.

Use cases don't make much sense when dealing with simple tasks or systems. There is no point in modeling a task flow of how a visitor can go to the "About us" page on a website. This is better done with site map diagrams. Use cases are primarily useful when dealing with more complex flows.

Use cases:

- Reflect functional requirements in an easy-to-read, easy-to-track text format.
- Represent the goal of an interaction between an actor and the system. (The goal represents a meaningful and measurable objective for the actor.)
- Record a set of paths (scenarios) that traverse an actor from a trigger event (start of the use case) to the goal (success scenarios).

- Record a set of scenarios that traverse an actor from a trigger event toward a goal but fall short of the goal (failure scenarios).
- Are multi-level—one use case can use/extend the functionality of another.

Use cases do not:

- Specify user interface design—they specify the intent, not the action detail.
- Specify implementation detail (unless it is of particular importance to the actor for assurance that the goal is properly met).

Use cases are used during many stages of software/system development to:

- Capture systems requirements.
- Act as a springboard for the software/system design.
- Validate the design.
- Validate proper and complete implementation of the use cases for test and quality assurance.
- Serve as an initial framework for the online help and user manual.

The use case model can be a powerful tool for controlling scope throughout a project's life cycle. Because a simplified use case model can be understood by all project participants, it can also serve as a framework for ongoing collaboration as well as a visual map of all agreed-upon functionality. It can, therefore, be a valuable reference during later negotiations that might affect the project's scope.

Use case models can be used to:

- Document the business process.
- Illuminate possible collaborative business areas.
- Separate business processes into functional system areas.
- Serve as requirements documentation for system development (because they are defined in a non-implementation/easy-to-read manner).
- Identify possibilities of system or component reuse.
- Categorize requirements (e.g., state of implementation, functional system).
- Rank requirements (e.g., level of importance, risk, level of interoperability).
- Publish requirements at various levels (e.g., detailed design requirements, hanger analysis requirements, document management, document creation requirements).
- Identify the effects of functional changes on implementation systems or of implementation changes on functional capabilities (because they are part of the object-oriented analysis and design process).

Best Practices and Lessons Learned

Validate use cases. Use cases are great for managing boundaries, exploring scope options, expanding scope, and controlling complexity, but they must be validated thoroughly. In reviewing use cases, ask these questions:

- Is the use case complete? Do any details need to be added?
- Do I feel confident that the actor's goal is going to be properly met?
- Can I suggest any procedural or requirement changes that would simplify the process depicted in the use case?
- Are there any additional goals of the actors that are not addressed?
- Are there any additional actors that are not represented (directly or indirectly)?

Have enough use cases. When do you know you have all the use cases or enough of them for your purposes? The simple answer is you have them all when the users, sponsor, and other stakeholders cannot think of any more.

Develop uses cases in a cooperative setting. Use cases can serve as a bridge between the needs of requirement analysts, designers, and system developers. But this will only happen if the different views of the participants are reflected in the use cases. Requirement analysts focus on client needs, interaction designers look at the operational user needs, and developers focus on the technological capabilities. If developed in a cooperative setting—and in context of budget and schedule—use cases can provide enduring utility.

Consider these common use case mistakes and pitfalls from Ellen Gottesdiener, a requirements expert and principal at EBG Consulting:

- Other important requirements representations are unused or underused.
- Use case goals are vague or lack clarity.
- Use case scope is ambiguous.
- Use case text includes nonfunctional requirements and user interface details.
- The initial use case diagrams excessively use "extends" and "includes."
- Use case shows inattention to business rule definition.
- Subject matter experts are not sufficiently involved in creating, reviewing, or verifying use cases.
- Preparation for user involvement in use case definition is insufficient.
- Too much detail appears too early in use case definition amid expectation that it is a one-pass activity.
- You failed to validate or verify your use cases.
- You have too few use cases. You've described only a subsystem or a few duties of a few subsystems. You've missed the greater, essential application.
- You have gone too far. Your use cases represent wishful thinking well outside of the needs of the operational user. This is a form of "creeping featurism" in which there is a danger of wasting time on features you're not keeping and then spending

more time to cull the list when you realize what has happened.

- You captured the wrong use cases. You've got plenty and could do without some, but you can't do without some you've missed.

Maintain a list of ways the system may be parameterized (vary the perceived importance of the actors, the types of input, future modes of operation, situations when good actors do bad things, exceptional circumstances). Iterate through your use cases to see if they cover all the situations. This might help refine your use cases or discover new ones.

Examine your nonfunctional requirements (e.g., constraints) to see if your use cases can address them. You might be able to refine, add, or drop use cases based on this.

Make a semantic network diagram (a quick brainstorm of all the concepts and interactions and relationships from the problem domain). Decide which concepts fall within the system (will be part of the object model), which are on the boundaries (probably will become actors or usages), and which are beyond the scope of the system being modeled (do not affect the software system being built and don't show up in any of the modeling efforts).

Don't make premature assumptions about the interface. Use cases should be thought of as abstract prototypes. The interface is yet to be designed, and premature assumptions can constrain the design unnecessarily. This is particularly important if you are not going to design the interface yourself.

Don't describe internal workings of the application/system. Use cases are tools for modeling user tasks; there are lots of other tools more suitable for system modeling.

Lessons from the Field

Stay focused on the business processes and not the information technology. Be clear that you and your customer agree on the issue or question being addressed. Focus on the users' challenges/issues. A multitude of use cases and mission threads can be pursued, but to do all of them would be unrealistic. Be sure to focus on and have objectives up front of the particular problem(s) that is being evaluated and solution(s) for that problem.

Be sure you have a firm grasp of the mission context. Thoroughly understand the end user's needs, concept of operations, and environment along with emerging threats, courses of actions, etc. (For example, don't do force-on-force now when perhaps anticipated future operations are insurgency oriented. Don't do quick shock and awe in isolation when potential is for protracted operations.) Focus on the end users' needs and not on what an operational leader wants—keep the use cases as close to reality as possible.

Exercise due diligence by looking for other definitions on the Web and elsewhere that are similar or related. Given the paucity of existing, documented use cases or mission threads, the first lesson learned is how to find them. The second is that no one considers anyone else's work authoritative.

Consider the varying environments as a key aspect of the use case/mission thread. This includes geographical/climate environments, cultural/social environments, and even military service or agency organization environments. Remember that although we strive for cross-agency, joint, or multinational considerations, the reality is that individual departments and agencies often drive many aspects of a use case; the mission thread and these drivers need to be considered.

Understand the level of operation that is being examined. Put the particular use case in context across the full spectrum of operations.

Aim to have some form of simulated results that can feed into the particular use cases you're examining. Simulate the input and output from the particular use case to and from the integrated set of activities involved in the particular operation or aspect of the execution. Use stimulation triggers to interplay with the particular use case(s) or mission threads that are under closer examination.

Extend the use case/thread perspective across the full suite of systems engineering activities: architectures, development, testing, training/exercises, and experiments. These will help move you beyond simply testing the buttons of a capability to testing/exercising a capability in the context of the end users' requirements.

Solicit tools to help with use case/mission thread creation, management, and use. There are a variety of commercial and government tools to help capture the use case actors, scripts, and actions. There are also tools to capture the results of these activities from modeling and simulations or operator-driven exercises/experiments. Remember, it is not about the tool but about the process.

Share use cases and mission threads with others. Given the relatively wide perspective and encompassing value of use cases and mission threads across the systems engineering life cycle, there is value in employing use cases/mission threads in multiple systems engineering activities.

Don't believe that you have the greatest use cases/mission threads in the world. You probably don't, but you probably do have important pieces of the overall puzzle and value; share use cases and mission threads to increase the overall value of capabilities to the end users.

An important part of using use cases and mission threads is the results you get from architectures, testing, experiments, and exercises. Be sure to make your results very visible. A lot of time is used in capturing and assessing the use case data and parameters. Make your analysis visible to others to use in exercises (perhaps physical exercises), models and simulations, etc. Seek out and contribute to use case/mission thread data repositories so your work can be used and reused across the systems engineering life cycle.

Remember to document. It is especially important to know who vetted your descriptions, what the scope of the analysis was, and who the user was. It is also important to provide references to any MITRE document that illuminates your work. Documentation is especially important when discussions occur later about scope.

References and Resources

Armour, F., and G. Miller, 2000, *Advanced Use Case Modeling: Software Systems*, Addison-Wesley Professional.

Bittner, K., and I. Spence, 2002, *Use Case Modeling*, Addison-Wesley Professional, pp. 2–3.

Carr, N., and T. Meehan, March 2, 2005, "Use Cases Part II: Taming Scope," A List Apart.

Cockburn, A., May 10, 2006, Use Case Fundamentals, http://alistair.cockburn.us/Use+case+fundamentals.

Cockburn, A., 2002, "Use cases, ten years later," http://alistair.cockburn.us/Use+cases%2c+ten+years+later, Originally published in *STQE* magazine, Mar/Apr 2002.

Cockburn, A., 2001, *Writing Effective Use Cases*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Denney, R., 2005, *Succeeding with Use Cases: Working Smart to Deliver Quality*, Addison-Wesley Professional.

Department of Defense, July 16, 2008, DoD Instruction 5200.39, Critical Program Information (CPI) Protection Within the Department of Defense.

Gottesdiener, E., June 2002, "Top Ten Ways Project Teams Misuse Use Cases—and How to Correct Them, Part I: Content and Style Issues," *The Rational Edge*.

Jacobson, I., 1992, *Object-Oriented Software Engineering*, Addison-Wesley Professional.

Liddle, S., 13 October 2009, Comment on "Stake holders for use case specifications," BYU Island, ISYS Core.

MITRE Lean Six Sigma Community of Practice, accessed December 2009.

Software Engineering Institute, "Evaluating System of Systems Architectures: Mission Thread Workshop."

Definition: *The Post-Implementation Review (PIR) is an evaluation tool that compares the conditions prior to the implementation of a project (as identified in the business case) with the actual results achieved by the project. With reference to information technology solution projects, the Information Systems Audit and Control Association (ISACA) defines the PIR as the first or subsequent review of an information technology (IT) solution and/or the process of its implementation, performed after its implementation [1].*

Keywords: *benefit analysis, evaluation, implementation review, investment management, performance management, post-technology use*

OTHER SE LIFE-CYCLE BUILDING BLOCKS ARTICLES

Acquiring and Incorporating Post-Fielding Operational Feedback into Future Developments: The Post-Implementation Review

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of a Post-Implementation Review (PIR) and the benefits and costs of employing them. They are expected to be able to recommend techniques for PIRs, assist the government in tailoring PIR procedures, lead PIRs, or perform individual PIR tasks, as appropriate (e.g., post-implementation technical performance analyses). However, since PIRs should be conducted by individuals not directly involved in the previous steps of the acquisition process, MITRE SEs are frequently precluded from participating in the reviews themselves, other than as subject matter experts, because of their role in recommending appropriate technology or providing guidance during earlier phases of the life cycle.

Background

The PIR is used to evaluate the effectiveness of system development after the system has been in production for a period of time. The objectives of the PIR are to determine if the system does what it is designed to do: Does it support the user as required in an effective and efficient manner? The review is intended to assess how successful the system is in terms of functionality, performance, and cost versus benefits, as well as to assess the effectiveness of the life-cycle development activities that produced the system. The review results can be used to strengthen the system as well as system development procedures. However, while the systems engineering community is in general agreement that the PIR is a laudable thing to do, in practice the review generally remains an ignored stepchild at best, as departments and agencies content themselves with simply "running the numbers" (i.e., doing statistical analyses of performance deltas).

Government Interest and Use

The President's Office of Management and Budget (OMB) outlines the purpose of the PIR in Step IV.3. Post-Implementation Review (PIR), of the Capital Programming Guide, first published in 1997, as a supplement to Circular A-11, Part 3: Planning, Budgeting, and Acquisition of Capital Assets. It is described as a diagnostic tool to evaluate the overall effectiveness of the agency's capital planning and acquisition process as a complement to operational analysis. This control mechanism is used during the operational life cycle of an asset to enable resource managers to optimize the performance of capital assets over the course of their life cycle and eventual disposition [2, 3]. OMB stipulates that PIRs should be conducted by individuals not directly involved in the acquisition of the asset, but that they may include owners and users of the asset or other personnel and consultants.

In response to the OMB mandate, agency system development life cycle (SDLC) documentation now specifies the preparation of a PIR report as part of the final phase of the SDLC, Operations and Maintenance. However, in its IT Investment Management Framework [4, 5], the General Accounting Office (GAO) observes that agencies will continue to have difficulty performing an effective PIR unless they have more comprehensively established policies and procedures to assess the benefits and performance of their investments. This lack of documented processes is typically one of GAO's primary criticisms of agency performance in this area (e.g., GAO's 2007 report on DoD Business System Modernization [GAO-07-538] which criticized the limited nature of the Defense Acquisition System [DAS] PIR procedures [6]). In its recently updated Acquisition Directive 102-01, Department of Homeland Security (DHS) requires a PIR of every program implemented in the agency. The most recent DoD Instruction 5000.02 extends the PIR requirement to acquisition category (ACAT) II and below [7].

According to OMB, PIRs should be conducted three to twelve months after an asset becomes operational. However, agencies vary in the length of that period, with a range of anywhere from three to eighteen months after implementation. GAO points out that the timing of a PIR can be problematic. A PIR conducted too soon after an investment has been implemented may fail to capture the full benefits of the new system, while a PIR conducted too late may not be able to draw adequately on the institutional memory of the development/investment process. GAO indicates that PIRs should also be conducted for initiatives that were aborted before completion, to assist in the identification of potential management and process improvements.

To ensure consistency of evaluations, OMB recommends the use of a documented methodology for the conduct of PIRs, requiring that the chosen methodology be in alignment with the organization's planning process. The required level of specificity for PIR reports varies from agency to agency. According to OMB, PIRs should address [2]:

Customer/User Satisfaction

- Partnership/involvement
- Business process support
- Investment performance
- Usage

Internal Business

- Project performance
- Infrastructure availability
- Standards and compliance
- Maintenance
- Security issues and internal controls
- Evaluations (accuracy, timeliness, adequacy of information)

Strategic Impact and Effectiveness

- System impact and effectiveness
- Alignment with mission goals
- Portfolio analysis and management
- Cost savings

Innovation

- Workforce competency
- Advanced technology use
- Methodology expertise
- Employee satisfaction/retention
- Program quality

In assisting government sponsors to tailor the PIR to a particular project, MITRE SEs should cast the net as broadly as possible, ensuring that the PIR also examines potential weaknesses and risks and the long-term maintainability of the solution.

Best Practices and Lessons Learned

In 2004, the Information Systems Audit and Control Association (ISACA) published a set of best practices specific to the PIR, the IS Auditing Guideline: Post-Implementation Review [1].

Properly leveraged, the PIR provides an important control mechanism and tool for the continual improvement of the acquisition process. However, PIR templates are all too frequently limited to checklists comparing baseline expectations to results on a numerical value scale or summary bullets rather than providing serious analyses of root causes and contributory factors. By contrast, the Systems Engineering: Post-Implementation Review (PIR) Document Template, published by the U.S. Army Information Technology Agency (ITA) as draft in October 2008, devotes the majority of its review to the system itself rather than to the acquisition process, and emphasizes that the review should result in a free-form report, thereby highlighting the analytical potential of the review [8].

That agencies should underestimate the value of PIRs is not surprising, given that managers frequently feel they are all too aware of the limitations of system releases as a result of compression of schedules and inevitable scope creep prior to deployment. The typical strategy is simply to "move on and fix the bugs" in the next release. "Why do we need a PIR if we already know what is wrong?," a rationale that is based on a failure to recognize the potential benefit of the review when it is conducted as a true audit of technical and process performance. Significantly, the PIR is not tied to any life-cycle milestone, effectively robbing it of any determinative value.

An additional reason for the failure to fully leverage the PIR as a process improvement tool for IT acquisitions may well be that because of the capital planning and investment control (CPIC) context in which OMB and GAO place the review, it is seen more as a capital management/project management tool than as a tool to improve system development from a technical perspective.

If, from a CPIC perspective, the PIR can be seen as the initial iteration of the operational assessments conducted over the life cycle of an operational system, from the SDLC perspective it is only loosely tied in to the systems life cycle, being relegated to the O&M phase rather than seen as a closeout report for the systems implementation phase of the life cycle. From the SE perspective, one could argue that the PIR is condemned to a technology limbo—too late for developers to care, too early for maintenance people to be interested.

If the PIR presents an excellent opportunity for MITRE SEs to assist their sponsors across all programs to improve the way they conduct business, this opportunity is even more acute for those MITRE programs whose major focus is providing systems engineering support to agencies involved in business systems modernization (BSM) efforts. These efforts generally encompass the entire enterprise and frequently, as in the case of DHS, involve the insertion of technologies that revolutionize the way the agencies do business and provide challenges far exceeding those faced in the implementation of systems in the past. The issues resulting from the challenges posed by BSM and major technology insertions are compounded by the fact that civilian agencies have increasingly relied on contractor support not just to deliver systems, but to provide end-to-end services. This is an environment where the PIR can be particularly useful not only from a project and portfolio management perspective but also as an analysis of the operational impact of new technology (e.g., the use of biometrically enabled passports to verify traveler identity or the use of digitized documentation in a traditionally paper-based environment).

The effectiveness of PIRs as a process improvement tool in general, particularly in the case of BSM deployments that transform the operating environment, is dependent on the willingness to address issues of workforce competence, advanced technology use, and methodology expertise (e.g., how database-driven and service-oriented architectures are received in organizations previously responsible for the maintenance of long-outdated hardware and software). Although these areas of investigation are identified in OMB's PIR guidance, they are generally absent from agency PIR procedures. However, it is frequently in these areas that implementations fail to achieve the expected results. This is either because the workforce is not ready to operate or maintain the new systems, those responsible for maintaining the new systems lack the expertise required to support next-generation technology delivered through the BSM program, or a mismatch exists between the advanced technology deployed and the requirements of the operational environment. If the PIR is not designed to explore the root cause of potential issues, the resulting report will remain a paper exercise. However, failure to confront difficult issues leaves agencies with limited tools in their dialog with oversight bodies when it comes to requesting additional resources or postponement of mandates.

References and Resources

1. ISACA, October 15, 2004, IS Auditing Guidelines Post-Implementation Review.
2. OMB, June 2006, "Post Implementation Review and Post-Occupancy Evaluation," Section III.3.3, Capital Programming Guide, Version 2.0, Supplement to Office of Management and Budget Circular A-11, Part 7: Planning, Budgeting, and Acquisition of Capital Assets.
3. OMB Circular No. A-130, February 8, 1996, Management of Federal Information Resources, revised.
4. GAO, 2004, Information Technology Investment Management. A Framework for Assessing and Improving Process Maturity (GAO-04-394G), pp. 83–89.

5. Internal Revenue Service, June 30, 2000, IRS Enterprise Life Cycle, Investment Decision Management, Post Implementation Review (PIR) Procedure.
6. GAO, May 2007, Business Systems Modernization: DoD Needs to Fully Define Policies and Procedures for Institutionally Managing Investments (GAO-07-538), p. 31.
7. Department of Defense, December 8, 2008, Department of Defense Instruction Number 5000.02.
8. U.S. Army Information Technology Agency, October 2008, ITA Systems Engineering. Post-Implementation Review (PIR) Document Template, Ver. 1.0.

Definition: *A system of systems (SoS) is "a collection of systems, each capable of independent operation, that interoperate together to achieve additional desired capabilities [1]." Test & Evaluation (T&E) is the process by which an SoS and/or its constituents are compared against capability requirements and specifications.*

Keywords: *system of systems (SoS), SoS test, SoS test and evaluation, SoS testing, SoS validation, SoS verification*

OTHER SE LIFE-CYCLE BUILDING BLOCKS ARTICLES

Test and Evaluation of Systems of Systems

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the characteristics of systems of systems (SoS) and the implications for systems engineering in an SoS environment, including SoS test and evaluation. SEs are expected to develop systems engineering and T&E plans for systems that are constituents of SoS as well as for SoS themselves.

Background

This article is a special subject that addresses unique aspects of T&E of SoS and outlines strategies and techniques for handling them.

Systems of systems (SoS) differ from traditional systems in a number of ways. As a result, the application of systems engineering to SoS requires that it be tailored to address the particular characteristics of SoS. Likewise, the distinctive characteristics of SoS have implications for the application of test and evaluation (T&E). This discussion specifically addresses "acknowledged SoS," a type of SoS that is growing in significance in the Department of Defense (DoD). Acknowledged SoS have recognized objectives, a designated manager, and resources. However, the constituent systems (those that interoperate with each other to achieve the SoS capabilities) retain their independent ownership, objectives, funding, development, and sustainment approaches. Changes in the constituent systems are based on collaboration between the SoS and the systems levels.

SoS raise unique development challenges consequent to the far-reaching SoS capability objectives: the lack of control by the SoS over the constituent systems and the dependence of SoS capability on leveraging already fielded systems that address user and SoS needs. Further, SoS are often not formal programs of record but rather depend on changes made through acquisition programs or operations and maintenance of fielded systems. As a result, the question addressed here is not simply, how do we implement T&E for SoS but rather, what does it mean to test and evaluate SoS?

SoS Characteristics Impacting Test and Evaluation

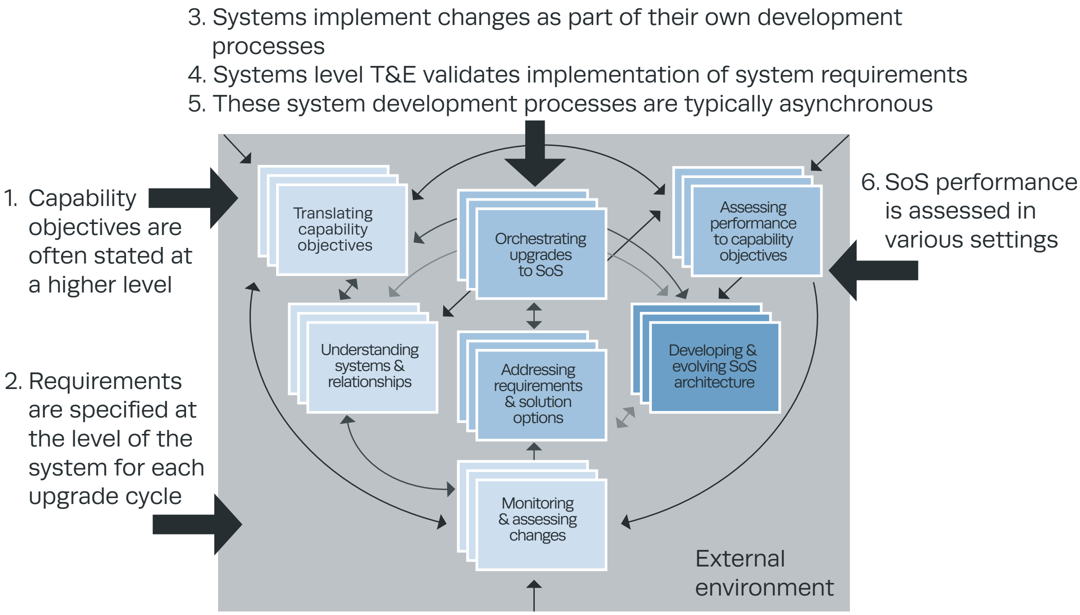
Table 1 summarizes key differentiating characteristics between systems and acknowledged SoS. Most of the differences are a result of the independence of the SoS's constituent systems. The constituent systems may evolve in response to user needs, technical direction, funding, and management control independent of the SoS. SoS evolution, then, is achieved through cooperation among the constituent systems, instead of direction from a central authority, by leveraging the constituent systems' efforts to improve their own individual capabilities.

An SoS will face T&E challenges that stem from the independence of its constituent systems:

- Independent development cycles mean that the delivery of systems' upgrades to meet SoS needs is done asynchronously and is bundled with other changes to the system in response to other needs (beyond those of the SoS).
- The number and variability of the systems that influence SoS results means that large SoS, in particular, are complex and that interactions among the constituents may lead to unintended effects or emergent behavior.

Table 1. Comparing Systems and Acknowledged Systems of Systems

Aspect of Environment	System	Acknowledged System of Systems
Management and Oversight		
Stakeholder Involvement	Clearer set of stakeholders	Stakeholders at both system level and SoS levels (including system owners), with competing interests and priorities; in some cases, the system stakeholder has no vested interest in the SoS; all stakeholders may not be recognized.
Governance	Aligned program manager and funding	Added levels of complexity due to management and funding for both the SoS and individual systems; SoS does not have authority over all the systems.
Operational Environment		
Operational Focus	Designed and developed to meet operational objectives	Called on to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives.
Implementation		
Acquisition	Aligned to acquisition category milestones, documented requirements, SE	Added complexity due to multiple system life cycles across acquisition programs involving legacy systems, systems under development, new developments, and technology insertion; Typically have stated capability objectives upfront which may need to be translated into formal requirements.
Test and Evaluation	Test and evaluation of the system is generally possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems' life cycles, given the complexity of all the moving parts and potential for unintended consequences.
Engineering & Design Considerations		
Boundaries and Interfaces	Focuses on boundaries and interfaces for the single system	Focus on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control, and functionality across the SoS while balancing needs of the systems.
Performance and Behavior	Performance of the system to meet specified objectives	Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems



3. Systems implement changes as part of their own development processes
4. Systems level T&E validates implementation of system requirements
5. These system development processes are typically asynchronous

1. Capability objectives are often stated at a higher level

2. Requirements are specified at the level of the system for each upgrade cycle

6. SoS performance is assessed in various settings

Translating capability objectives
Orchestrating upgrades to SoS
Assessing performance to capability objectives
Understanding systems & relationships
Addressing requirements & solution options
Developing & evolving SoS architecture
Monitoring & assessing changes
External environment

Figure 1. SoS SE Core Elements and Their Relationships to T&E

Test and Evaluation in the SoS SE Process

The SoS Guide [1] presents seven core SoS SE elements. Four are critical to T&E of the SoS. More detail on these elements can be found in the above reference; this article summarizes their key aspects as shown in Figure 1. The discussion that follows shows how T&E activities fit into the SoS SE core elements and the challenges SoS pose for T&E.

1. **Capability objectives of an SoS are often stated at a high level, particularly when the need for an SoS is first established.**

   Translating capability objectives into high-level SoS requirements is a core element in the SoS SE process. In most cases, SoS capability objectives are framed in high-level language that needs to be interpreted into high-level requirements to serve as the foundation of the engineering process.

   "SoS objectives are typically couched in terms of needed capabilities, and the SE is responsible for working with the SoS manager and users to translate these into high-level requirements that provide the foundation for the technical planning to evolve the capability over time [1, p. 18]."

   These objectives establish the capability context for the SoS, which grounds the assessment of current SoS performance. In most cases, SoS do not have requirements per se; they have capability objectives or goals that provide the starting point for specific requirements that drive changes in the constituent systems to create increments of SoS evolution.

2. **Requirements are specified at the level of the system for each SoS upgrade cycle.**

In the SoS SE core element, "assessing requirements and solution options," increments of SoS improvements are planned collaboratively by managers and SEs at the SoS and system levels. Typically, there are specific expectations for each increment about system changes that will produce an anticipated overall effect on the SoS performance. While it may be possible to confidently define specifications for the system changes, it is more difficult to do this for the SoS, which is, in effect, the cumulative result of the changes in the systems.

"It is key for the systems engineer to understand the individual systems and their technical and organizational context and constraints when identifying viable options to address SoS needs and to consider the impact of these options at the systems level. It is the SoS systems engineer's role to work with requirements managers for the individual systems to identify the specific requirements to be addressed by appropriate systems (that is to collaboratively derive, decompose, and allocate requirements to systems) [1, p. 20]."

As a result, most SoS requirements are specified at the system level for each upgrade cycle, which provides the basis for assessing system-level performance. As discussed below, T&E of system changes is typically done by the systems as part of their processes.

3. **Systems implement changes as part of their own development processes.**

The main source of T&E challenges arises from SoS upgrades that are the product of changes in independent operating systems and in the SoS itself. The SoS SE team needs to work with the SE systems teams to plan and track these systems changes that will contribute to meeting the SoS capability objectives:

"Once an option for addressing a need has been selected, it is the SoS systems engineer's role to work with the SoS sponsor, the SoS manager, the constituent systems' sponsors, managers, SEs, and contractors to fund, plan, contractually enable, facilitate, integrate, and test upgrades to the SoS. The actual changes are made by the consistent systems' owners, but the SoS systems engineer orchestrates the process, taking a lead role in the synchronization, integration, and test across the SoS and providing oversight to ensure that the changes agreed to by the systems are implemented in a way that supports the SoS [1, p. 20]."

4. **Systems-level T&E validates implementation of system requirements.**

Consequently T&E is implemented as part of this element at both the system and SoS levels. It seems fairly straightforward to assess whether the systems have made the changes as specified in the plan; however, it is less clear how the results of these changes at the SoS level are to be tested and evaluated.

"Throughout orchestration, the systems are implementing changes according to the negotiated plans, and they are following their own SE and T&E processes. The SoS systems engineer works with the SE teams of the constituent systems to enable SoS insight into progress of the system developments as laid out in the SoS plan. The SoS SE team members are responsible for integration and for verification and validation of the changes across the suite of system updates under an SoS increment, including T&E tailored to the specific needs of the increments. Their efforts may result in both performance assessments and a statement of capabilities and limitations of the increment of SoS capability from the perspectives of SoS users and users of the individual systems. These assessments may be done in a variety of venues, including distributed simulation environments, system integration laboratories, and field environments. The assessments can take a variety of forms, including analysis, demonstration, and inspection. Often SoS SEs leverage system-level activities that are underway in order to address SoS issues [1, p. 68]."

There are significant challenges in creating an end-to-end test environment sufficient to addressing the met needs of the SoS capability. This can be mitigated by conducting T&E on a subset of systems prior to fielding the entire SoS increment, though at the expense of some T&E validity. Contingency plans should be prepared for when the SoS T&E results don't reflect expected improvements in case the systems are ready to be fielded based on system-level test results and owners' needs.

5. **Constituent system development processes are typically asynchronous.**

The asynchronous nature of the constituent systems development schedules presents a challenge to straightforward T&E at the SoS level. While it is obviously desirable to coordinate the development plans of the systems and synchronize the delivery of upgrades, as a practical matter, this is often difficult or impossible. Even when it is possible to plan synchronous developments, the result may still be asynchronous deliveries due to the inevitable issues that lead to development schedule delays, particularly with a large number of systems or when the developments are complex.

"SoS SE approaches based on multiple, small increments offer a more effective way to structure SoS evolution. Big-bang implementations typically will not work in this environment; it is not feasible with asynchronous independent programs. Specifically, a number of SoS initiatives have adopted what could be termed a 'bus stop,' spin, or block-with-wave type of development approach. In this type of approach, there are regular time-based SoS 'delivery' points, and systems target their changes for these points. Integration, test, and evaluation are done for each drop. If systems miss a delivery point because of technical or programmatic issues, they know that they have another opportunity at the next point (there will be another bus coming to pick up passengers in 3 months, for instance). The impact of missing the scheduled bus can be evaluated and addressed. By providing this type of SoS battle rhythm, discipline can be inserted into the inherently asynchronous SoS environment. In a complex SoS environment, multiple iterations of incremental development may be underway concurrently.

"Approaches such as this may have a negative impact on certification testing, especially if the item is software related to interoperability and/or safety issues (such as Air Worthiness Release). When synchronization is critical, considerations such as this may require large sections of the SoS, or the entire SoS, to be tested together before any of the pieces are fielded [1, pp. 68–69]. "

As these passages indicate, the asynchronous nature of system developments frequently results in other SoS constituents being unprepared to test with earlier delivering systems, complicating end-to-end testing. However, as autonomous entities, constituent systems expect to field based on results of their own independent testing apart from the larger impact on SoS capability. Holding some systems back until all are ready to test successfully is impractical and undesirable in most cases. These dependencies form a core impediment to mapping traditional T&E to SoS.

6. **SoS performance is assessed in various settings.**

SoS typically have broad capability objectives rather than specific performance requirements as is usually the case with independent systems. These capability objectives provide the basis for identifying systems as candidate constituents of an SoS, developing an SoS architecture, and recommending changes or additions to constituent systems.

"In an SoS environment there may be a variety of approaches to addressing objectives. This means that the SoS systems engineer needs to establish metrics and methods for assessing performance of the SoS capabilities which are independent of alternative implementation approaches. A part of effective mission capability assessment is to identify the most important mission threads and focus the assessment effort on end-to-end performance. Because SoS often comprise fielded suites of systems, feedback on SoS performance may be based on operational experience and issues arising from operational settings, including live exercises as well as actual operations. By monitoring performance in the field or in exercise settings, SEs can proactively identify and assess areas needing attention, emergent behavior in the SoS, and impacts on the SoS of changes in constituent systems [1, pp. 18–19]."

This suggests the necessity of generating metrics that define end-to-end SoS capabilities for ongoing benchmarking of SoS development. Developing these metrics and collecting data to assess the state of the SoS is part of the SoS SE core element "assessing the extent to which SoS performance meets capability objectives over time." This element provides the capability metrics for the SoS, which may be collected from a variety of settings as input on performance, including new operational conditions [1, p. 43]. Hence, assessing

SoS performance is an ongoing activity that goes beyond assessment of specific changes in elements of the SoS.

T&E objectives, particularly key performance parameters, are used as the basis for making a fielding decision. In addition, SoS metrics, as discussed above, provide an ongoing benchmark for SoS development which, when assessed over time, should show an improvement in meeting user capability objectives. Because SoS typically comprise a mix of fielded systems and new developments, there may not be a discrete SoS fielding decision; instead, the various systems are deployed as they are ready, at some point reaching the threshold that enables the new SoS capability.

In some circumstances, the SoS capability objectives can be effectively modeled in simulation environments that can be used to identify changes at the system levels. If the fidelity of the simulation is sufficient, it may provide validation of the system changes needed to enable SoS-level capability. In those cases, the fidelity of the simulation may also be able to provide for the SoS evaluation.

In cases where simulation is not practical, other analytical approaches may be used for T&E. Test conditions that validate the analysis must be carefully chosen to balance test preparation and logistics constraints against the need to demonstrate the objective capability under realistic operational conditions.

Best Practices and Lessons Learned

Approach SoS T&E as an evidence-based approach to addressing risk. Full conventional T&E before fielding may be impractical for incremental changes to SoS because systems may have asynchronous development paths. In addition, explicit test conditions at the SoS level may not be feasible due to the difficulty in bringing all constituent systems together to set up meaningful test conditions. Thus an incremental risk-based approach to identifying key T&E issues is recommended.

For each increment, a risk-based approach identifies areas critical to success and areas that could have adverse impacts on user missions. This is followed by a pre-deployment T&E. Risk is assessed using evidence from a range of sources, including live test. In some circumstances, the evidence can be based on activity at the SoS level, in others it may be based on roll-ups of activity at the constituent systems level. The activity can range from explicit verification testing, results of models and simulations, use of linked integration facilities, and results of system level operational test and evaluation.

Finally, these risks must be factored into SoS and system development plans, in case the T&E results indicate that the changes will have a negative impact, which can then be discarded without jeopardizing system update deliveries to users. The results could be used to provide end-user feedback in the form of "capabilities and limitations."

Encourage development of analytic methods to support planning and assessment. Analytical models of the SoS can serve as effective tools to assess system-level performance against SoS operational scenarios. They may also be used to validate the requirements allocations to systems, and provide an analytical framework for SoS-level capability verification. Such models may used to develop reasonable expectations for SoS performance. Relevant operational conditions should be developed with end user input and guided by "design of experiments" principles, so as to explore a broad a range of conditions.

Address independent evaluation of networks that support multiple SoS. Based on the government vision of enabling distributed net-centric operations, the "network" has assumed a central role as a unique constituent of every SoS. Realistic assessment of SoS performance demands evaluation of both network performance and potential for degradation under changing operational conditions. Because government departments and agencies seek to develop a set of network capabilities for a wide range of applications, consideration should be given to developing an approach to network assessment independent of particular SoS applications as part of SoS planning and T&E.

Employ a range of venues to assess SoS performance over time. For SoS, evaluation criteria may be end user metrics that assess the results of loosely defined capabilities. While these may not be expressly timed to the development and fielding of system changes to address SoS capability objectives, this data can support periodic assessments of evolving capability and provide valuable insight to developers and users.

Assessment opportunities should be both planned and spontaneous. For spontaneous opportunities, T&E needs to be organized in a way that facilitates responding flexibly as they arise.

Establish a robust process for feedback once fielded. Once deployed, continuing evaluation of the fielded SoS can be used to identify operational problems and make improvements. This continuous evaluation can be facilitated through system instrumentation and data collection to provide feedback on constraints, incipient failures warnings, and unique operational conditions.

By establishing and exercising robust feedback mechanisms among field organizations and their operations and the SoS SE and management teams, SoS T&E can provide a critical link to the ongoing operational needs of the SoS. Feedback mechanisms include technical and organizational dimensions. An example of the former is instrumenting systems for feedback post-fielding. An example of the latter is posting a member of the SoS SE and management team to the SoS operational organization.

References and Resources

1. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), August 2008, Systems Engineering Guide for Systems of Systems, Washington, DC.

Additional References and Resources

Dahmann, J., G. Rebovich, J. Lane, R. Lowry, and J. Palmer, 2010, "Systems of Systems Test and Evaluation Challenges," 5th IEEE International Conference on System of Systems Engineering.

The MITRE Institute, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model, Ver. 1," Section 2.6.

Definitions: *Verification is the process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications [1].*

Validation is the process of determining the degree to which a simulation model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model [1].

Keywords: *accreditation, analysis, data, historical, SME, statistical, subject matter expert, validation, verification*

OTHER SE LIFE-CYCLE BUILDING BLOCKS ARTICLES

Verification and Validation of Simulation Models

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to have a sound knowledge of the system being modeled and the software process for developing the model in order to provide effective technical guidance in the design and execution of plans to verify and/or validate a model, or to provide specialized technical expertise in the collection and analysis of varying types of data required to do so. They are expected to be able to work directly with the developer of the system and the simulation model to provide technical insight into model verification and validation. In most cases, MITRE SEs will be responsible for assisting the government sponsoring organization in the formal accreditation of the model.

Background

Modeling and simulation (M&S) can be an important element in the acquisition of systems within government organizations. M&S is used during development to explore the design trade space and inform design decisions, and in conjunction with testing and analysis to gain confidence that the design implementation is performing as expected, or to assist troubleshooting if it is not. M&S allows decision makers and stakeholders to quantify certain aspects of performance during the system development phase, and to provide supplementary data during the testing phase of system acquisition. More important, M&S may play a key role in the qualification ("sell-off") of a system as a means to reduce the cost of a verification test program. Here, the development of a simulation model that has undergone a formal verification, validation, and accreditation (VV&A) process is not only desirable, but essential.

Table 1. Common Simulation Model Validation Methods

Model Validation Method	Description
Comparison to Other Models	Various results (e.g., outputs) of the simulation model being validated are compared to results of other (valid) models. For example, (1) simple cases of a simulation model are compared to known results of analytic models and (2) the simulation model is compared to other simulation models that have been validated.
Face Validity	Asking individuals knowledgeable about the system whether the model and/or its behavior are reasonable. For example, is the logic in the conceptual model correct and are the model's input-output relationships reasonable?
Historical Data Validation	If historical data exist (e.g., data collected on a system specifically for building and testing a model), part of the data are used to build the model and the remaining data are used to determine (test) whether the model behaves as the system does.
Parameter Variability – Sensitivity Analysis	This technique consists of changing the values of the input and internal parameters of a model to determine the effect on the model's behavior of output. The same relations should occur in the model as in the real system. This technique can be used qualitatively—directions only of outputs—and quantitatively—both directions and (precise) magnitudes of outputs. Those parameters that are sensitive (i.e., cause significant changes in the model's behavior or output) should be made sufficiently accurate prior to using the model.
Predictive Validation	The model is used to predict (forecast) the system's behavior, and then the system's behavior and the model's forecast are compared to determine if they are the same. The system's data may come from an operational system or be obtained by conducting experiments on the system, e.g., field tests.

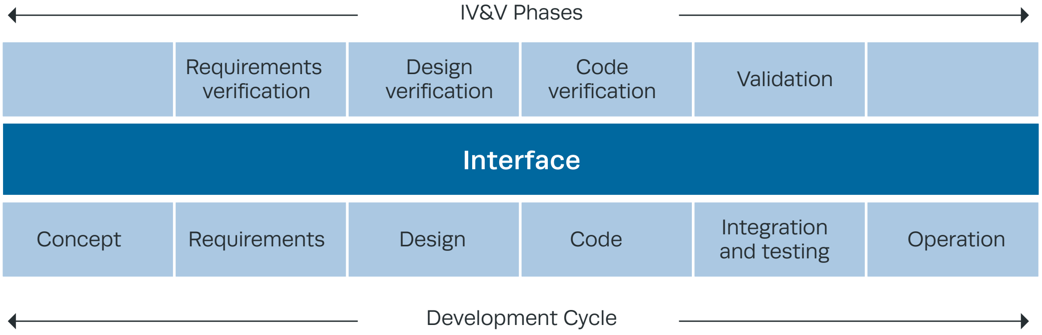


Figure 1. The Relationship Between IV&V and Systems Engineering Processes

Consider the following definitions for the phases of the simulation model VV&A process [1]:

- **Verification:** "The process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications."
- **Validation:** "The process of determining the degree to which a [simulation] model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model."
- **Accreditation:** "The official certification that a model, simulation, or federation of models and simulations and its associated data are acceptable for use for a specific purpose."
- **Simulation Conceptual Model:** "The developer's description of what the model or simulation will represent, the assumptions limiting those representations, and other capabilities needed to satisfy the user's requirements."

Verification answers the question "Have we built the model right?" whereas validation answers the question "Have we built the right model?" [2]. In other words, the verification phase of VV&A focuses on comparing the elements of a simulation model of the system with the description of what the requirements and capabilities of the model were to be. Verification is an iterative process aimed at determining whether the product of each step in the development of the simulation model fulfills all the requirements levied on it by the previous step and is internally complete, consistent, and correct enough to support the next phase [3]. The validation phase of VV&A focuses on comparing the observed behavior of elements of a system with the corresponding elements of a simulation model of the system, and on determining whether the differences are acceptable given the intended use of the model. If agreement is not obtained, the model is adjusted in order to bring it in closer agreement with the observed behavior of the actual system (or errors in observation/experimentation or reference models/analyses are identified and rectified).

Typically, government sponsoring organizations that mandate the use of a formal VV&A process do not specify how each phase should be carried out. Rather, they provide broad guidance that often includes artifacts required from the process. Independent verification and validation (IV&V) activities occur throughout most of the systems engineering development life-cycle phases and are actively connected to them, as depicted in Figure 1, rather than being limited to integration and testing phases.

A variety of methods are used to validate simulation models, ranging from comparison to other models to the use of data generated by the actual system (i.e., predictive validation). The most commonly used methods are described in Table 1 [4].

With the exception of face validity, all the methods detailed in Table 1 are data-driven approaches to model validation, with predictive validation among the most commonly used methods. The use of predictive validation generally requires a significant amount of effort to acquire and analyze data to support model validation.

Best Practices and Lessons Learned

Develop and maintain a model VV&A plan. Develop a detailed model VV&A plan before the start of the acquisition program VV&A process. Distinct from the specification for the model itself, this plan provides guidance for each phase of VV&A and clarifies the difference between the verification and validation phases. The plan should also map model specification requirements to model elements, identify those model elements that require validation, and develop model validation requirements. The plan should describe the method(s) used for validation, including any supporting analysis techniques. If a data-driven approach is used for model validation, detail should be provided on either the pedigree of existing data or the plan to collect new data to support validation. Prototype simulations and ad hoc models are often developed where a VV&A plan is not considered beforehand. Then, when the prototype becomes a production system, an attempt is made to perform V&V. It is extremely difficult to perform verification after the fact when normal system development artifacts have not been created and there is no audit trail from concept to product.

Establish quantitative model performance requirements. Often performance requirements are neglected while developing the domain model. Complex systems can have interactions that produce unexpected results in seemingly benign situations. Prototypes developed with small problem sets may not scale to large problems that production systems will deal with. More model detail does not necessarily generate a better answer and may make a simulation intractable. In discrete event simulation, the appropriate event queue implementation, random number generators, and sorting/searching algorithms can make a huge difference in performance.

Establish quantitative model validation requirements. Specify the degree to which each element (component, parameter) of the model is to be validated. For model elements based on a stochastic process, validation is often based on statistical tests of agreement between the behavior of the model and that of the actual system. If a hypothesis test is used to assess agreement, both allowable Type I and Type II risks (errors) need to be specified as part of an unambiguous and complete validation requirement. If, instead, a confidence interval is used to assess agreement, the maximum allowable interval width (i.e., precision) and a confidence level need to be specified.

In certain instances, the decision to use either a hypothesis test or a confidence interval may be a matter of preference; however, there will be instances when using a hypothesis test is necessary, due to the nature of the model element being validated. Regardless, the development of a validation requirement is often an involved task requiring the use of analytic or computational methods to determine allowable levels of validation risk or precision. Sufficient time, resources, and expertise should be allocated to this task.

Develop model validation "trade-off" curves. If predictive validation is used, the amount of data required to achieve a quantitative model validation requirement (see above) will need to be determined. Determining the amount of data required (N) to achieve, say, a maximum allowable confidence interval width at a specified confidence level for any model element may require extensive analytical or computational methods. Related to this is solving the "inverse" problem: determining the maximum (or expected) confidence interval width given a particular value of N. From this, a set of curves may be constructed to allow for the "trade-off" between the amount of validation data required (which, generally, drives the cost of the validation effort) in order to achieve a validation requirement. This problem should be addressed as early as possible during the validation phase of the VV&A process.

Not every model element requires validation. Model elements associated with model functional requirements usually do not require validation. These elements are only dealt with in the verification phase of the VV&A process. Trivial examples of this are model elements that allow the user to select various model options (i.e., "switches" or "knobs"). Nontrivial examples are elements that have been identified as not being relevant or critical to the intended use of the model. However, a model element that has not been deemed critical may, in fact, be fully functional when the simulation model is deployed. In this case, the model element may still be exercised, but the model accreditation documentation should note that this particular element has been "verified, but not validated."

Allow for simplifications in the model. Almost always, some observed behaviors of the actual system will be difficult to model or validate given the scope and resources of the model development and validation efforts. In such cases, using simplifications (or approximations) in the model may provide an acceptable way forward. For example, if a model element requires a stochastic data-generating mechanism, a probability density

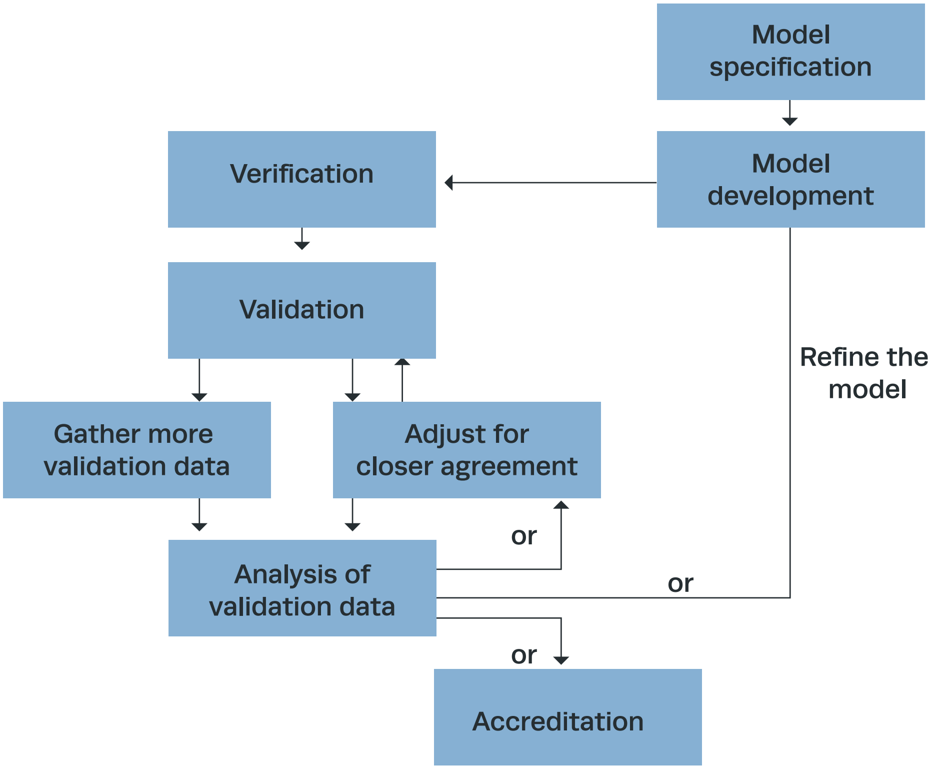


Figure 2. Simulation Model Development and the VV&A Process

function with a limited number of parameters (e.g., a Gaussian distribution) may be used in place of what appears to be, based on analysis of data from the actual system, a more complex data-generating mechanism. In doing this, a conservative approach should be used. That is, in this example, employing a simplified data-generating mechanism in the model should not result in overly optimistic behavior with respect to the actual system.

Plan for parallel (iterative) model development and VV&A. The model development and its VV&A process should be carried out in parallel, with sufficient resources and schedule to allow for several iterations. Figure 1 depicts a notional relationship between model development and the VV&A process. The focus here is the gathering and analysis of validation data for a particular model element and the resulting decision to: (1) adjust the value of one or more parameters associated with the model element to obtain closer agreement with observed system behavior, (2) redesign the model element by factoring in insights obtained from the analysis of validation data, or (3) accredit the model with respect to this element.

Although Figure 1 depicts the relationship between model development and VV&A in its entirety, it may be applied to individual, independent model elements. A model VV&A plan should identify those model elements that may be validated independently of others.

Consider the partial model validation of selected model elements. When the validation of a particular model element becomes problematic, it may be acceptable to validate the model element over a subset of the element's defined range. This partial validation of a model element may be a viable option when either insufficient data is available to enable validation or the actual system behavior is difficult to model even if reasonable simplifications are made. However, the resulting model will be valid with respect to this element only within this limited range. This fact should be noted in accreditation documentation.

Use multiple approaches to model validation. When data-driven model validation is not possible or practical, face validity may be used. However, even when data-driven validation can be carried out, face validity (i.e., expert review) may be used as a first step in the validation phase. If a similar but already validated model is available, performing a comparison of the model being developed to this existing model may provide an initial (or preliminary) model validation. Following this initial validation, a more comprehensive approach such as predictive validation may be employed.

Subsystem verification and validation do not guarantee system credibility. Each submodel may produce valid results and the integration of models can be verified to be correct, but the simulation can still produce invalid results. Most often this occurs when the subsystem conceptual design includes factors that are not considered in the system conceptual design, or vice versa. For example, in a recently reviewed simulation system with multiple subcomponents, one of the subcomponents simulated maintenance on vehicles, including 2½ ton trucks. There was no need for the submodel to distinguish between truck types; however, vehicle type distinction was important to another module. In the system, tanker trucks used for hauling water or petroleum were being returned from the maintenance submodel and became ambulances—alphabetically the first type of 2½ ton trucks.

Know what your simulation tool is doing. If a simulation language or tool is used to build the simulation, "hidden" assumptions are built into the tool. Here are four common situations that are handled differently by different simulation tools:

- **Resource recapture:** Suppose a part releases a machine (resource) and then immediately tries to recapture the machine for further processing. Should a more highly qualified part capture the machine instead? Some tools assign the machine to the next user without considering the original part a contender. Some tools defer choosing the next user until the releasing entity becomes a contender. Still others reassign the releasing part without considering other contenders.
- **Condition delayed entities:** Consider two entities waiting because no units of

a resource are available. The first entity requires two units of the resource, and the second entity requires one unit of the resource. When one unit of the resource becomes available, how is it assigned? Some tools assign the resource to the second entity. Other tools assign it to the first entity, which continues to wait for another unit of the resource.

- **Yielding control temporarily:** Suppose that an active entity wants to yield control to another entity that can perform some processing, but then wants to become active and continue processing before the simulation clock advances. Some tools do not allow an entity to resume processing at all. Other tools allow an entity to compete with other entities that want to process. Still others give priority to the relinquishing process and allow it to resume.
- **Conditions involving the clock:** Suppose an entity needs to wait for a compound condition involving the clock (e.g., "wait until the input buffer is empty or it is exactly 5:00 p.m."). Generally the programmer will have to "trick" the system to combine timed events with other conditions. An event at 5:00 p.m. could check to see if the buffer was empty, and if so, assume that the buffer-empty event occurred earlier.

Test for discrete math issues. Computer simulation models use discrete representations for numbers. This can cause strange behaviors. When testing models, always include tests at the extremes. Examples of errors found during testing include overflow for integers and subtraction of floating point numbers with insufficient mantissa representation. Conversion from decimal to binary representation could cause rounding errors that are significant in the simulation.

Establish a model validation working group. A regular and structured working group involving the sponsoring government organization, the system developer, and the model developer will result in a superior simulation model. The function of this group should be the generation and review of artifacts from the model validation phase. When data-driven validation is used, the majority of the effort should focus on the information products produced by the statistical analysis of validation data. This group may also provide recommendations regarding the need to collect additional validation data should increased quality in the validation results be necessary.

Invest in analysis capabilities and resources. Plans for the availability of subject matter experts, sufficient computing and data storage capability, and analysis software should be made early in the VV&A process. For many VV&A efforts, at least two subject matter experts will be required: one who has knowledge of the system being modeled and another who has broad knowledge in areas of data reduction, statistical analysis, and, possibly, a variety of operations research techniques. If analysis software is needed, consider using open-source packages—many provide all the data reduction, statistical analysis, and graphical capability needed to support validation efforts.

Summary

The successful validation of a simulation model requires the government sponsoring organization to address VV&A early in the life of an acquisition program. Key activities include:

- The development of a model VV&A plan with quantitative model validation requirements (where appropriate).
- Detailed planning for the collection of validation data (if data-driven validation is needed).
- The assembly of a working group that includes both domain experts and analysts.

Model development and model validation should not be carried out sequentially, but in a parallel and iterative manner.

References and Resources

1. DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A), December 9, 2009, DoD Instruction 5000.61.
2. Cook, D. A., and Skinner, J. M., May 2005, How to Perform Credible Verification, Validation, and Accreditation for Modeling and Simulation, *CrossTalk: The Journal of Defense Software Engineering*.
3. Lewis, R. O., 1992, *Independent Verification and Validation*, Wiley & Sons.
4. Law, A. M., 2008, How to Build Valid and Credible Simulation Models, paper presented at the Winter Simulation Conference.

Additional References and Resources

Law, A. M., 2007, *Simulation Modeling and Analysis*, McGraw Hill.

Winter Simulation Conference, http://wintersim.org/2014/

OTHER SE LIFE-CYCLE BUILDING BLOCKS ARTICLES

Affordability, Efficiency, and Effectiveness (AEE)

Definitions: *Affordability, Efficiency, and Effectiveness (AEE) are three success measures that guide systems engineers in developing and shaping engineering solutions, making program recommendations, and evaluating engineering efforts.*

Keywords: *analysis of alternatives, budgets, cost benefit analysis, life-cycle cost, portfolio analysis, program cost, return on investment*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to incorporate and assess measures of affordability, efficiency, and effectiveness in engineering solutions and supporting acquisition activities of sponsors. They are expected to:

- Work with users to understand their mission needs, capability gaps, and performance requirements.
- Develop alternative solutions or courses of action, and evaluate them for mission effectiveness as well as life-cycle affordability and efficiency.
- Understand operational and technical domains; recommend and conduct engineering-based trade-offs of requirements, design, performance, cost, and schedule to address affordability constraints.

- Understand life-cycle cost, schedule, risk, and affordability implications of alternatives under consideration, and incorporate these dimensions in engineering products and recommendations.
- Encourage and facilitate active participation of the broad stakeholder community and acquisition decision makers in exploring alternatives; help them understand and use the trade space to achieve program affordability and evaluate merits of alternatives from operational as well as business perspectives.
- Monitor and evaluate contractor system development efforts; identify affordability risks and recommend changes when warranted as an acquisition program progresses.
- Assist government sponsors in developing, adjusting, and implementing strategies, at program and enterprise levels, to ensure affordability and improve efficiency/effectiveness.
- Communicate AEE best practices and lessons learned.

Background

The measures of affordability, efficiency, and effectiveness can be characterized as follows:

- **Affordability:** Ability to fund desired investment. Solutions are affordable if they can be deployed in sufficient quantity to meet mission needs within the (likely) available budget.
- **Efficiency:** A measure of the “bang for the buck” or “unit benefit per dollar.” Solutions are efficient if they measurably increase the “bang” or “unit benefit” for the amount of resources required to deliver the capability.
- **Effectiveness:** “The bang”; the ability to achieve an organization’s mission. Solutions are effective if they deliver capability of high value to accomplishing the user’s missions [1].

MITRE’s mission is to work in partnership with its government sponsors in applying science and advanced technology to engineer systems of critical national importance. MITRE’s systems engineering is conducted in the context of developing solutions to meet the needs and challenges of our sponsors in conducting their missions, and in aiding sponsors in planning and managing programs to acquire such solutions. Sponsor success is achieved if the systems they deploy are effective and available when needed to achieve their mission. Systems to be procured, deployed, and sustained must be affordable, that is, within the means of the sponsor’s available resources, and should be efficient, providing high value for the resources to be expended.

Sponsor acquisition environments present many challenges. A number of system development programs have failed to deliver needed capabilities, or delivered reduced capability with expenditure of time and funds well beyond what was planned. Current mounting federal budget deficits place considerable economic stress on government agencies. Budget reductions mandate difficult decisions about where to invest limited resources, how to make current programs more affordable, and whether to terminate poorly performing programs. Investments for new capabilities, replacements, or enhancements to existing systems as well as simple continuation of existing programs require careful analysis and evaluation of their affordability, efficiency, and effectiveness. Systems engineering seeks to apply current and emerging technologies flexibly to address sponsors' dynamic threats and mission needs. At the same time, *affordability engineering* helps sponsors respond to fiscal realities and be effective stewards of taxpayer dollars.

As depicted in Figure 1, affordability challenges exist at different sponsor levels and are addressed by varying engineering, analysis, or management approaches.

Enterprises invest in and sustain systems, infrastructures, and organizations to accomplish multiple missions. Investment decisions made by agency heads, Senior Acquisition Executives (SAEs), Chief Information Officers (CIOs), and Program Executive Officers (PEOs) require a holistic view of the enterprise's missions and objectives. Investment decisions are required for new or enhanced systems that provide additional capabilities as well as for sustainment of existing systems, infrastructures, manpower, and operations. A portfolio management approach evaluates the benefits of or return on investment choices relative to filling

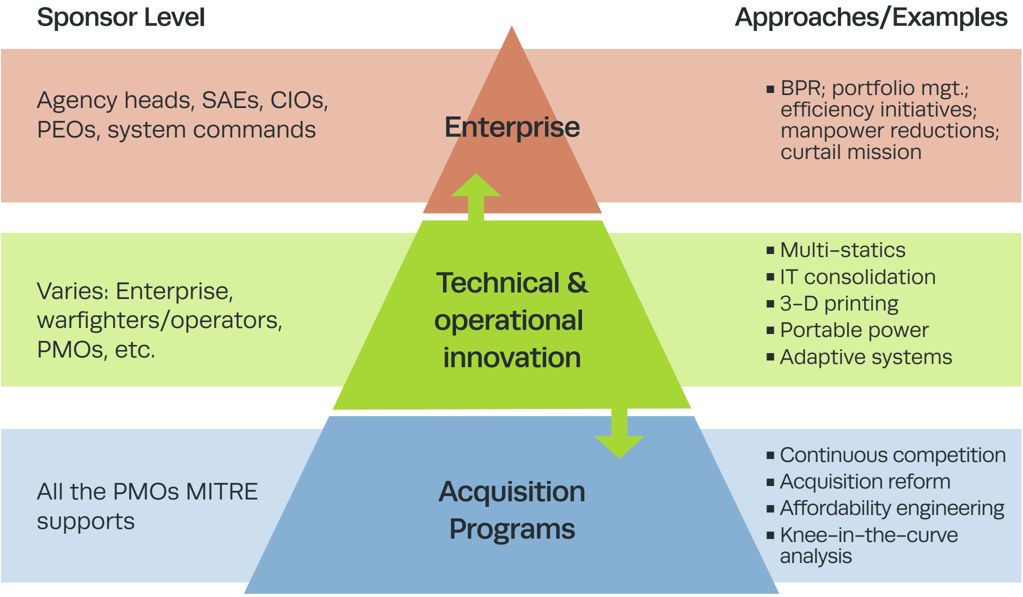

Sponsor Level
Approaches/Examples
Agency heads, SAEs, CIOs, PEOs, system commands
Enterprise
▪ BPR; portfolio mgt.; efficiency initiatives; manpower reductions; curtail mission
Varies: Enterprise, warfighters/operators, PMOs, etc.
Technical & operational innovation
▪ Multi-statics
▪ IT consolidation
▪ 3-D printing
▪ Portable power
▪ Adaptive systems
All the PMOs MITRE supports
Acquisition Programs
▪ Continuous competition
▪ Acquisition reform
▪ Affordability engineering
▪ Knee-in-the-curve analysis

Figure 1. AEE Construct

identified capability gaps or improving mission functions. Efficiency initiatives and business process reengineering (BPR) look to reduce redundancy, overlap, or inefficient processes. Enterprise-level analysis and decision frameworks can be applied to help an agency achieve greatest mission effectiveness, ensuring highest priority/highest value needs are met with its allocated budget [1].

Each acquisition program is an element of the integrated capability delivered by the enterprise. As such, it is vital that each be executed to deliver its target capability within available technology, funding, and time (i.e., be affordable). Each program and capability must contribute high value to users in terms of mission effectiveness achieved by the most efficient means. Not doing so can, like a chain reaction, have serious impacts across the enterprise. At this level, engineering solutions for affordability and adopting best acquisition systems engineering and management practices within the Program Management Office (PMO) are key to achieving success.

Technical and operational innovation contributes to AEE at both the acquisition program and enterprise levels. Adaptive systems and composable capabilities provide an enterprise with the flexibility to respond to rapidly changing mission needs with existing resources and minimal new investment. Application of advances in IT and network design offer potential for great efficiency in the delivery of information and services to end users at the enterprise and program level. Advances within technology domains afford an opportunity to reduce life-cycle costs in the acquisition of new or enhanced capabilities and to transform operations and sustainment approaches for greater efficiency.

Government Interest and Use

The U.S. economy is experiencing an era of very slow growth, high unemployment, historic high debt, and mounting budget deficits at federal and lower levels. Considerable economic stress is being felt by all government agencies as they strive to accomplish their missions and deliver services with constant or shrinking budgets. Agencies across the federal government are implementing new strategies to promote AEE in their acquisition decisions and management practices.

In June 2010, the Office of the Secretary of Defense/Acquisition, Technology and Logistics (OSD/ATL) memo Better Buying Power (BBP): Mandate for Restoring Affordability and Productivity in Defense Spending set an “important priority” for DoD: “delivering better value to the taxpayer and improving the way the Department does business” [2]. OSD/ATL memo Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending, September 14, 2010, followed, setting the significance and breadth of this mandate [3]. This 17-page memo outlined five key areas, with more than 20 specific initiatives. The five areas are:

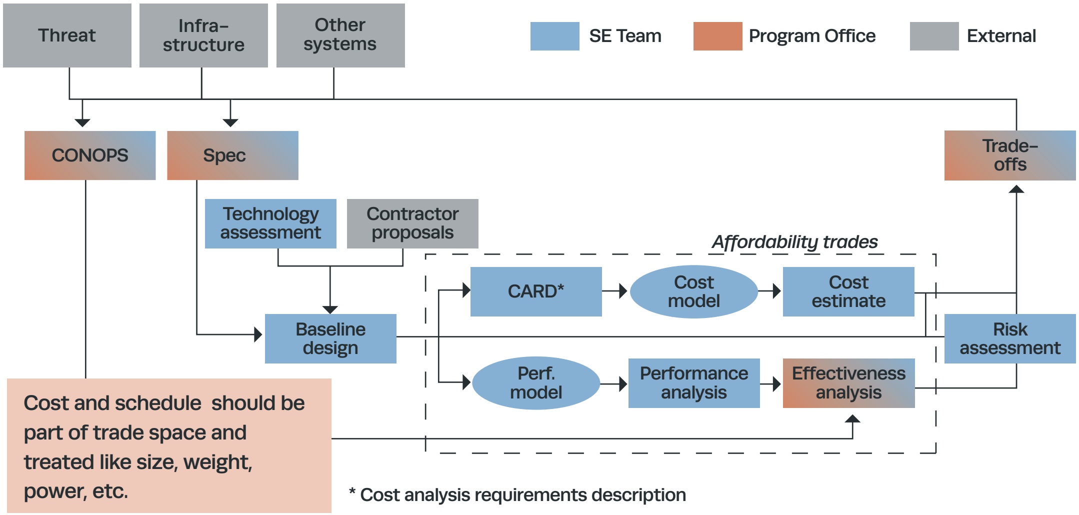


Figure 2. Affordability in Systems Engineering [6]

1. Target Affordability and Control Cost Growth (mandating affordability as a requirement and implementing "should-cost" based management)
2. Incentivize Productivity and Innovation in Industry
3. Promote Real Competition
4. Improve Tradecraft in Services Acquisition
5. Reduce Non-Productive Processes and Bureaucracy

AT&L has issued further guidance with specific requirements and management practices to address affordability in program planning and execution and milestone decisions. In response, acquisition leadership in the various services has also issued implementation directives. The Secretary of Defense has set targets for cost, budget, and personnel reductions in many areas of DoD's operations as well. Additional references on the topic of affordability are available in Section 3 of "Affordability Engineering Capstone (Phase I) Volume 1 - Basic Research" [4].

The Office of Management and Budget and the General Accounting Office (GAO) have also been targeting AEE in government spending, addressing acquisition and contracting practices, duplicative capabilities and services, and inefficient business operations [4]. Although the practice of AEE in acquisition is not new (acquisition guidance and regulation pre-BBP have long been concerned with delivering capabilities within cost and schedule targets), what is new is the sense of urgency given the current economic crisis.

Achieving AEE

At program levels, achieving AEE requires constant effort across the acquisition life cycle to examine cost and schedule implications of choices. Decisions about the system design, what requirements to meet, and how to structure and manage the acquisition impact affordability and introduce potential affordability risk. As illustrated in Figure 2, cost and schedule analysis is integral to the systems engineering process. It will reveal affordability risks and define the trade space of mission needs, cost, schedule, and performance in which to explore and critically evaluate alternatives. Engineering and cost analysis must be closely coupled in the process. Systems engineering and technical skills must be paired with cost estimating skills to examine affordability trades and recommend analysis-based courses of action.

As the program moves through its acquisition stages, divergence of the technical baseline and cost estimate must be carefully monitored. Early discovery of divergence permits early intervention and correction, reducing affordability risk.

An Affordability Engineering Risk Evaluation (AERiE) tool [5] is being developed to facilitate identification of affordability risk at multiple points across the acquisition life cycle. AERiE is also envisioned as part of an Affordability Engineering Framework (AEF) that will facilitate the validation of the technical baseline and program cost estimate, suggest and analyze trade-offs to address affordability disconnects, and recommend alternative courses of action.

From several studies, the GAO identified proven acquisition practices to minimize the risk of cost growth on DoD programs. Such practices help "establish programs in which there is a match between requirements and resources—including funding—from the start and execute those programs using knowledge-based acquisition practices [7]." Although referring to DoD, these practices are generally applicable to acquisition programs of any government agency. They require a strong systems engineering foundation be established early in a program and greater reliance on a government systems engineering team to set and manage objectives. Practices include:

- **Early and continued systems engineering analysis:** Ideally beginning before a program is initiated, early systems engineering is critical to designing a system that meets requirements (or negotiates requirements) within available resources, such as technologies, time, money, and people. A robust analysis of alternatives and a preliminary design review (PDR)—which analyze the achievability of required capabilities before committing to a program—can help ensure that new programs have a sound, executable business case that represents a cost-effective solution to meeting the critical user needs. Such engineering knowledge can identify key trade-offs in requirements and technology that are essential to managing cost. Systems engineering continues to be an important tool through a program's critical design review (CDR) and system demonstration.

- **Leveraging mature technologies and processes:** Programs often have insufficient knowledge about the maturity of technology. Prototyping early in programs can provide confidence that a system's proposed design can meet performance requirements. Further, having predictable manufacturing processes before decisions are made to move into production can reduce unknowns. Naturally this assumes that the manufacturing process is used to develop the prototype.
- **Establishing realistic cost and schedule estimates matched to available resources:** Cost and schedule estimates are often based on overly optimistic assumptions. Without the ability to generate reliable cost estimates, programs are at risk of experiencing cost overruns, missed deadlines, and performance shortfalls. Inaccurate estimates do not provide the necessary foundation for sufficient funding commitments. Engineering knowledge and more rigorous technical baselines are required to achieve more accurate, reliable cost estimates at the outset of a program. Established cost estimating, schedule estimating, work-breakdown structures, risk management techniques, engineering analyses, and past performance help achieve realism in AEE assessments.
- **Clear, well-defined requirements:** Government department and agency cultures and environments sometimes allow programs to start with too many unknowns, for example, entering the acquisition process without a full understanding of requirements (technical, training, integration, fielding environment, etc.). Minimizing requirements changes could decrease the amount of cost growth experienced by acquisition programs, but this has to be carefully managed and balanced in an evolving environment to ensure continued effectiveness against, for example, new or improved adversary threats.
- **Incremental approach to acquiring capabilities:** Programs can put themselves in a better position to succeed by implementing incremental/evolutionary acquisition strategies that limit the time in each incremental development.

At enterprise or portfolio levels, a number of analyses and approaches are applied to assess affordability and promote efficiency and effectiveness in investment decisions. Each is appropriate to a decision or management context. MITRE SEs are expected to understand key aspects of the analyses that will need to be performed. They are expected to know the objectives of the analysis, the decisions to be supported, and the general approaches that can be applied. They are expected to enlist the support of and engage with analysts in conducting analyses supporting AEE objectives.

SEs are frequently called on to perform or support a number of different investment analysis types (analysis of alternatives, business case analysis, and cost benefit analysis—to name a few). These are focused on informing sponsor funding and expenditure decisions and they provide critical analysis for assessing affordability, efficiency, and effectiveness of alternatives in deciding to select a solution or course of action.

- **Analysis of Alternatives (AoA):** An AoA is a technical assessment using distinct metrics and different criteria to objectively evaluate different potential courses of action (or alternatives). Typically the emphasis is focused on an analysis of alternative technical approaches, measuring their effectiveness in meeting a given set of functional requirements or mission need. The AoA also includes a life-cycle cost estimate for each alternative, a risk assessment for each alternative, and a recommendation(s) regarding a preferred alternative, pending the results of a more rigorous business case analysis.
- **Business Case Analysis (BCA):** A BCA is used to determine if a new approach and overall acquisition should be undertaken. A BCA results in a justification, one way or the other, based on the comparison of life-cycle costs and benefits and the results of financial analysis techniques such as return on investment (ROI), net present value (NPV), and payback for each alternative. A BCA may evaluate a single or multiple alternatives against the status quo. Based on the results of the financial analysis, a BCA will help to determine if a potential new acquisition is warranted and if the effort should go forward.
- **Cost Benefit Analysis (CBA):** A cost benefit analysis is a structured assessment of alternative courses of action for achieving some objective. A CBA looks forward and evaluates specific courses of action to determine which would yield the maximum ROI. The assessment informs a decision maker about financial, non-financial, and other non-quantifiable impacts—costs and benefits—of each course of action.

These analyses are described in more detail in the articles “Performing Analyses of Alternatives” and “Comparison of Investment Analyses.”

Best Practices and Lessons Learned

AEE is not achieved through the application of any single analytic approach or engineering or management practice, or even a small set of the same. AEE practices need to be integrated throughout enterprise and program engineering and acquisition management activities. Achieving AEE of acquisition programs or in enterprise operations requires a continual conscious effort on the part of all stakeholders. The following practices are fundamental to engineering for AEE and achieving successful acquisitions. They reflect some examples of best practices and are derived from l essons learned:

Understand the operational mission, its context, and the current systems or solutions employed. Understand what is changing, and what is influencing these changes. What do these changes imply in terms of new operational needs?

As an engineer, understand the current program architecture and system operations to be able to evaluate impacts of these changes. Also understand the principles of the enterprise architecture, the data and system interdependencies, and required interoperability. Affordability considerations extend beyond the system boundaries. This understanding can be gained through discussions with end users and participation in operational exercises and experiments.

Understand the operational gaps, mission deficiencies, or enhanced/new capabilities being sought by users. What are the users' imperatives (threat, time, consequences) to meet these needs? Determine required vs. desired capabilities and performance levels. At what performance level would an improved capability provide no substantive value beyond current capabilities? At what performance level would an improved capability exceed that required to accomplish the mission? Resources spent delivering performance in excess of that needed might be more effectively applied to other needs. The understanding of operational gaps can be gained through examining the after-action assessments of operations, various operational lessons learned, etc.

Derive solutions from consideration of DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership, Personnel, Facilities) alternatives, not just material solutions. Where can non-material solutions affect the desired enhanced capabilities and operational benefit? If a material solution is deemed necessary, determine the non-material changes also needed to achieve the desired capability. Are these changes accounted for in program plans and life-cycle cost estimates? Understanding up front the full DOTMLPF impact of a solution is key to avoiding affordability surprises later in the program.

Conduct market research to determine where exploiting or adapting commercial products or services in devising solutions may be possible. Understand the product marketplace, product maturity, and the business as well as the technical/operational and logistics risks of reliance on commercial or government products. Many technology and capability assessments as well as product reviews exist and can help. Reach out to others via social media.

Assess the value proposition. From a portfolio point of view, evaluate the cost-effectiveness of solutions compared to alternative expenditures of available resources on other needs or capabilities. Is the expenditure of resources "worth it?" Does the enhanced or new capability provide value to users higher than addressing other important needs? Engineering assessments highlighted below (e.g., analysis of alternatives) provide techniques for evaluating the value proposition.

Use early systems engineering to define the trade space in which alternatives can be developed and evaluated. Define multiple concepts and characterize them technically with sufficient information to support rough order of magnitude cost estimation. Use concept modeling, modeling and simulation, prototyping, or experimentation to examine concept feasibility. Identify the cost and schedule drivers of the concepts as they relate to specific requirements. Involve system users to identify technical or performance requirements that can be traded off to achieve cost and schedule objectives, or to define what capabilities can be affordably delivered. Identify the requirements that drive cost and/or schedule and that impose greater risk to timely delivery of needed capabilities. Work with the users and other stakeholders as needed to define evolutionary approaches to meeting these requirements.

Assess and compare the life-cycle cost, effectiveness, and risks of alternatives in selecting a solution. Ensure decision processes drive efficient and effective solution choices. Measure the affordability of each solution against a current budget profile and assess the affordability risk if the budget is changed. Understand and use established cost estimating tools to help determine cost drivers and major risks associated with the AEE of a capability. (See the article "Life-Cycle Cost Estimation.")

Assess user stakeholder expectations against realism of budgets, time, and technology maturity. Understand the basis of budgets and funding profiles. Ensure they are consistent with the chosen solution/technical approach, based on a cost estimate of a suitable technical baseline, and include assessment of cost and schedule risk. Be wary of downward-directed schedules. Develop engineering-based timelines showing the critical paths and dependencies; ensure that risks and uncertainty have been incorporated. For developmental items, ensure that a technology readiness assessment (see the article "Assessing Technical Maturity") accurately characterizes the technology maturity, and that the effort and time to advance maturity to achieve desired performance or other requirements are adequately assessed. Present the realism in cost as well as operational terms of what mission aspects will be and might not be totally satisfied by the recommended approach along with the feasibility/projection of capability satisfaction over time/future evolutions to help stakeholders assess trade-offs. Create a time-phased roadmap highlighting the recommended AEE strategy for implementation of capabilities.

Establish, document, and maintain a comprehensive, stable technical baseline to support timely cost analysis and design trades. The technical baseline of a chosen solution becomes the foundation for the program cost estimate and program planning and execution. Through program implementation, it serves as the basis for performance of design and strategy trade-offs, risk management, and mitigation analyses. For these purposes, the technical baseline must provide a holistic description of the system that includes its technical and functional composition, its relationships and interdependencies with other elements of the enterprise, and its acquisition strategy and program implementation.

Communicate the technical baseline to ensure cost analysts understand it. Work with the cost analysts in developing a comprehensive work breakdown structure that captures all aspects of the technical baseline. Provide a credible engineering basis and make clear any assumptions regarding input to the technical baseline. Ensure that stakeholders—user community, acquisition community, oversight organizations, etc.—are aware of, familiar with, and understand the trade-offs of the technical baseline and its role in AEE.

Assess the completeness and realism of the program's cost and schedule estimate. Consider the program's alignment and completeness with respect to the technical baseline and any changes to it as well as the adequacy with which uncertainty and risk have been integrated. As system requirements and program strategies change, the technical baseline as well as the program cost estimate should be updated.

Integrate management of cost and technical baselines throughout the program. Ensure that cost, engineering, and management teams work together (ideally collocated) to keep the technical baseline and Program Cost Estimate current, and maintain a list of risks, cost drivers, and alternative COAs/mitigations to address moderate/high-risk areas.

Treat cost and schedule as part of the design-capabilities trade space, just like size, weight, power, security, throughput, and other engineering parameters. Understand user expectations/targets for total system cost, particularly unit procurement and sustainment costs for systems with large quantities to be installed or fielded. Assess the ability of the chosen design to meet these targets.

Understand and document all system interfaces, interoperability requirements, dependencies on other systems, programs, and resources, and assess their associated risk as it would impact the program. The interfaces and dependencies of capabilities from independent, yet associated, efforts can be a big contributor to cost due to schedule mismatches, reworking of misunderstood interface exchanges, increased complexity in testing, etc. Include consideration of these tasks and dependencies in the technical and cost baselines along with the operational utility/value of the interfaces, dependencies, and interoperability. Various crown jewel and map-to-mission techniques can be used to help accomplish this. These techniques are frequently used for cyber mission assurance assessments and are equally valuable to these AEE analyses.

Manage affordability as a key risk parameter in the contractor's system development effort. Use periodic design reviews to ensure that each component of the system is on track from a risk perspective (technical, cost, and schedule) to meet functional, performance, and interface requirements. Monitor design change for impacts to production and sustainment costs.

Inform key design and programmatic decisions with assessment and understanding of affordability implications and associated risks. Maintain and measure progress against AEE objectives (metrics) in design, engineering, and management reviews and decision processes. Ensure "affordability" is communicated to decision makers. Conduct independent assessments when confronted with significant change in affordability risk.

Keep users well informed and involved in major engineering decisions affecting requirements satisfaction, trade-offs, and affordability. Present the AEE risks (as highlighted earlier in "Achieving AEE") to the user community for their decisions in accepting the risks (e.g., increased costs balanced against increased effectiveness) to achieve an overall best value solution.

References and Resources

1. Affordability, Efficiency, and Effectiveness, Corporate Brief, December 2011, The MITRE Corporation.
2. OSD/AT&L Memorandum for Defense Acquisition and Logistics Professionals, Better Buying Power: Mandate for Restoring Affordability and Productivity in Defense Spending, June 28, 2010.
3. OSD/AT&L Memorandum for Defense Acquisition and Logistics Professionals, Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending, September 14, 2010.
4. MITRE Working note WN110058V1 "Affordability Engineering Capstone (Phase I) Volume 1 - Basic Research," J. Duquette et al., September 2011.
5. AERiE prototype tool Community Share site.
6. GAO-11-318SP Opportunities to Reduce Potential Duplication in Government Programs, Save Tax Dollars, and Enhance Revenue, March 2011.
7. GAO Testimony before the Committee on Homeland Security and Governmental Affairs, Subcommittee on Federal Financial Management, Government Information, Federal Services and International Security, United States Senate. DOD COST OVERRUNS Trends in Nunn-McCurdy Breaches and Tools to Manage Weapon Systems Acquisition Costs, March 29, 2011, pp. 6–8.

Acquisition Systems Engineering

MITRE

Introduction

MITRE systems engineers (SEs) perform systems engineering activities in various contexts. This includes support to field users, operational headquarters, acquisition agencies and program offices, policy and oversight organizations, as well as independent efforts of all forms (e.g., red teams, blue teams) across a range of collaborating stakeholders, such as other Federally Funded Research and Development Centers (FFRDCs), industry, and academia. SEs are expected to adapt systems engineering principles to these different environments.

MITRE SEs are expected to work with the government customer to plan and manage the overall FFRDC systems engineering activities to support government acquisition efforts. They plan and technically manage MITRE systems engineering efforts, and sometimes those of others, for projects and programs throughout the system life cycle. For systems engineering activities that cut across multiple phases of a government program or activity, MITRE SEs are expected to make connections among them and plan and manage their execution.

The focus of this section is on applying MITRE systems engineering support to government acquisition programs. Most of the topics and articles are directed exclusively to the subject of acquisition. The others apply to systems engineering in support of government activities that include but go beyond acquisition.

Background

Many processes and models have emerged to guide the execution of systems engineering activities in government acquisition processes. Over time, important best practice themes have emerged. They can be found throughout this section and include:

- **Planning and management are iterative.** Management is sometimes thought of as execution, although a more common view is that management is both planning and execution. Although planning is always done early in an activity, many of the best practices and lessons learned of this section have a common element of iteration. For example, during execution of a planned acquisition, conditions often change, calling for a change in the plan as well. One of the topics that addresses this is Continuous Process Improvement.
- **Risk management is the first job that needs to be done.** Some program managers characterize program and project management as risk management because once a program or project is started, the work of the SEs and managers often focuses on identifying potential problems and their solutions. The Risk Management topic in this section provides guidance on this aspect of systems engineering. The risk identification and mitigation theme also appears in other SEG articles, for example, “Competitive Prototyping” and “Systems Engineering Strategies for Uncertainty and Complexity.”

- **Think "enterprise."** Most programs being developed or undergoing significant modifications are already interfacing to a number of other systems, networks, databases, and data sources over the Web, or are part of a family or system of systems. Explore the Enterprise Engineering section to gain more knowledge on how to approach systems engineering for these cases, and fold that thinking into the acquisition systems engineering efforts you are undertaking. For example, the Transformation Planning and Organizational Change topic explores the criticality of stakeholders as champions of program success.

The topics addressed in this section are summarized below.

Acquisition Program Planning

Acquisition is the conceptualization, initiation, design, development, test, contracting, production, deployment, logistic support, modification, and disposal of systems, supplies, products, or services (including construction) to satisfy agency/department needs, intended for use in or in support of that organization's mission.

Acquisition program planning is concerned with the acquisition-related coordination and integration efforts undertaken to meet agency or department needs. The scope and type of the systems engineering support MITRE provides is determined by where along the spectrum—from purchasing commodity to major development—the supported acquisition effort falls. MITRE SEs are expected to understand the central role that systems engineering plays in effectively managing acquisition programs (or projects). Because systems engineering and program management are so inextricably linked, MITRE SEs need to be cognizant of program management challenges and issues. MITRE SEs may be required to assist in planning the technical work; create, staff, and direct a team or organization to do the work; monitor progress against the plan; and take corrective action to control and redirect the work when needed. Articles in this topic area include "Performing Analyses of Alternatives," "Acquisition Management Metrics," "Assessing Technical Maturity," "Technology Planning," "Life-Cycle Cost Estimation," "Integrated Master Schedule (IMS)/Integrated Master Plan (IMP) Application," and "Comparison of Investment Analyses."

Source Selection Preparation and Evaluation

The purpose of source selection is to prepare for a government solicitation, evaluate responses to the solicitation, and select one or more contractors for delivery of a product or service. MITRE SEs are expected to create technical and engineering portions of request for proposal (RFP) documentation (requirements documents, statement of work, evaluation criteria) and to assist in the technical evaluation of bidders. The technical evaluation is an assessment of the degree to which proposed solutions or courses of action will provide the capabilities required to meet the government's needs. This role includes conducting assessments of risk inherent in proposed solutions, including strategies for acquiring (or implementing) them, and identifying actionable options for mitigating those risks. Articles in this topic area include "Picking the Right Contractor" and "RFP Preparation and Source Selection."

Program Acquisition Strategy Formulation

An acquisition strategy is a comprehensive, integrated plan developed as part of acquisition planning activities. It describes the business, technical, and support strategies to meet program objectives and manage program risks. The strategy guides acquisition program execution across the entire program (or system) life cycle. It defines the relationship among the acquisition phases and work efforts as well as key program events such as decision points, reviews, contract awards, test activities, production lot/delivery quantities, and operational deployment objectives. The strategy evolves over time and should continuously reflect the current status and desired end point of the program. MITRE SEs assist in articulating government needs, translating those needs into mission/outcome-oriented procurement/solicitation requirements, and identifying the issues, risks, and opportunities that shape and influence the soundness of the acquisition strategy. MITRE SEs help agencies achieve what the Federal Acquisition Regulation (FAR) characterizes as "mission-oriented solicitations" (FAR 34.005-2). Articles in this topic area include "Agile Acquisition Strategy," "Evolutionary Acquisition," and "'Big-Bang' Acquisition."

Contractor Evaluation

Contractor evaluation assesses the contractor's technical and programmatic progress, approaches, and deliverables. The purpose of contractor evaluation is to provide insight into risks and the likelihood of meeting program and contractual requirements. MITRE SEs perform contractor evaluations and milestone reviews, influence sponsor/customer decisions during those reviews, monitor the contractor's continued performance, and recommend changes based on their performance. This topic is related to the MITRE FFRDC Independent Assessments topic in the Enterprise Engineering section and contributes significantly to the process of identifying and managing risks, as discussed in the Risk Management topic. Articles in this topic area include "Data-Driven Contractor Evaluations and Milestone Reviews," "Earned Value Management," and "Competitive Prototyping."

Risk Management

Defining and executing an iterative risk management process is a significant component of effective acquisitions and programs. MITRE SEs propose the risk management approach that enables risk-informed trade-offs and decisions to be made throughout a system's evolution, and they are actively involved in all steps of the process. Articles in this topic area include “Risk Management Approach and Plan,” “Risk Identification,” “Risk Impact Assessment and Prioritization,” “Risk Mitigation Planning, Implementation, and “Progress Monitoring,” and “Risk Management Tools.”

Configuration Management

Configuration management (CM) is the application of sound program practices to establish and maintain consistency of a product or system’s attributes with its requirements and evolving technical baseline over its lifetime. Configuration management is required by the Department of Defense (DoD), Federal Aviation Administration (FAA), Internal Revenue Service (IRS), and other programs that MITRE supports. MITRE SEs assist in ensuring that good CM processes are in place and followed by all contractors and program office personnel. Articles in this topic area include “How to Control a Moving Baseline” and “Configuration Management Tools.”

Integrated Logistics Support

Integrated logistics support is the management and technical process through which supportability and logistic support considerations are integrated into the design and taken into account throughout the life cycle of systems and equipment. MITRE SEs need to understand the impact of technical decisions on the usability and life-cycle support of systems and assist in ensuring that life-cycle logistics considerations are part of the SE process. Articles in this topic area include “Reliability, Availability, and Maintainability” and “Managing Energy Efficiency.”

Quality Assurance and Measurement

Quality assurance (QA) and measurement are systematic means for ensuring that defined standards and methods are applied. The rigorous application of quality assurance and measurement is one mechanism to mitigate program risk. MITRE SEs recommend and assist in executing QA and measurement programs. Articles in this topic area include “Establishing a Quality Assurance Program in the Systems Acquisition or Government Operational Organization” and “How to Conduct Process and Product Reviews Across Boundaries.”

Continuous Process Improvement

Continuous process improvement is an aspect of quality assurance. It is the set of ongoing systems engineering and management activities used to select, tailor, implement, and assess the processes used to achieve an organization’s business goals. MITRE SEs influence the government’s approach to implementing and improving systems engineering processes. Articles in this topic area include “Implementing and Improving Systems Engineering Processes for the Acquisition Organization” and “Matching Systems Engineering Process Improvement Frameworks/Solutions with Customer Needs.”

Other Acquisition Systems Engineering Articles

In the future, any articles on subjects of relevance to enterprise engineering but that don’t neatly fit under one of the section’s existing topics will be added in a separate topic, Other Acquisition Systems Engineering Articles. Such articles are likely to arise because the subject matter is at the edge of our understanding of systems engineering, represents some of the most difficult problems MITRE SEs work on, and has not yet formed a sufficient critical mass to constitute a separate topic.

References and Resources

General Services Administration, Department of Defense, National Aeronautics and Space Administration, March 2005, Federal Acquisition Regulation, Vol. 1, 34.005-2.

The MITRE Corporation, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, ver. 1.1E, section 3.0, pp. 34–48.

Acquisition Systems Engineering Contents

Acquisition Program Planning

Definition: *Acquisition planning is the process for coordinating and integrating acquisition efforts by using a plan to fulfill agency needs in a timely manner and at a reasonable cost. It includes developing the overall strategy for managing the acquisition. Planning enables the coordinated execution of the various efforts that constitute acquisition management [1, part 2].*

Keywords: *acquiring capabilities, acquisition, acquisition management, analysis of alternatives, contracting, information technology, program management, scope of acquisition, software engineering, systems engineering*

Context

MITRE systems engineers (SEs) support a wide spectrum of federal acquisition management efforts through its federally funded research and development centers (FFRDCs). Government acquisition efforts range from purchasing commodities (which MITRE is not involved in), such as commercially available goods and services requiring little or no modification or system integration, to developing major, unprecedented systems that provide new strategic capabilities to resource the various dimensions of U.S. national security.

These federal agencies/departments each have governance frameworks for managing acquisitions through contracts. Though there is a "family resemblance" across the frameworks, there are differences as well. Some (such as DoDI 5000.02 [2]) have evolved over many years and are fairly mature, whereas other frameworks are in their infancy. Federal Acquisitions Regulations (FAR) provides guidance to agencies on policies and procedures for implementing FAR requirements. This includes conducting acquisition planning and execution to ensure that the government acquires systems and capabilities in an effective, economical, and timely manner [1, subpart 1.3, part 7]. However, the FAR does not specify a particular acquisition management or governance framework. Instead, it provides the latitude to put in place an acquisition planning system appropriate for that agency/department [1, subpart 7.102]. The acquisition systems adopted by these departments and agencies are very diverse and require MITRE SEs to adapt systems engineering concepts, principles, and processes to align with the supported agency's acquisition processes and governance/management framework.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to understand the central role systems engineering plays in effectively planning and managing government acquisition programs to acquire products, systems, capabilities, and services to satisfy mission and business needs or modernize enterprises. MITRE SEs are expected to tailor and adapt systems engineering principles, processes, and concepts to match the scope and complexity of the acquisition effort as well as the agency or department acquisition regulations, policies, and governance approaches. They need to be cognizant of program management challenges and issues so they can assume appropriate accountability for the success of the programs they support. SEs may be required to plan the technical work; create, staff, and guide a team or organization to do the work; monitor progress against the plan; and advise the government regarding corrective action to control and redirect the work when needed [3].

Best Practices and Lessons Learned

Help your customer take a holistic perspective. MITRE experience addressing the challenges of developing, operating, and sustaining complex systems highlights the need to take a broad view of acquisition, with systems engineering as an integral part of it to be applied across the acquisition spectrum or life cycle.

This holistic perspective of acquisition has taken hold in information technology (IT)-intensive acquisitions (e.g., capital investments) that are part of enterprise modernization initiatives managed under the Capital Planning and Investment Control framework, a component of the Clinger-Cohen Act [4] that contains several provisions for improving the way agencies acquire IT. OMB Circular A-11 [5] provides additional guidance and instructions for the planning, budgeting, acquisition, and management of IT assets.

MITRE SEs should articulate the role systems engineering can take in improving the outcomes of IT acquisition programs. Several key areas where systems engineering plays a major role in achieving desired outcomes are discussed in the articles under this topic, along with suggestions on how to address the challenges typically encountered.

Know the customer's acquisition governance framework. Because the maturity of agencies' acquisition governance frameworks differs, MITRE SEs should be aware of their agency's level of maturity and tailor systems engineering practices to the acquisition environment of the supported agency.

Make use of MITRE's collected experience. Provide value by drawing on lessons learned from similar MITRE efforts. Look for common problems and potentially common solutions that can be applied across multiple programs and agencies. This involves conversing in various terminologies to express similar concepts or ideas. In other cases, the same terminology may take on a different meaning. MITRE SEs should articulate recommendations, such as courses of actions to address key technical risks, in the language of the supported agency.

Acquisition management metrics: Outcome. MITRE SEs should ensure that acquisition planning addresses needs using outcome-focused attributes and associated metrics. Questions such as these should be asked and answered:

- What are the success criteria for the acquisition effort? How will we know when we are done?
- What mission or business shortfall (or gap) is the acquisition effort trying to address? Can the need (shortfall/gap) be characterized as attributes that broadly define the solution space (i.e., range of potential alternatives for satisfying the need)?
- Are the attributes/metrics used to articulate the need traceable to criteria established during analysis of alternative solution concepts (e.g., types of investments) for addressing the need?

Outcome-focused metrics provide the information needed for program management and systems engineering activities such as risk and trade space management, test and verification/evaluation, modeling and simulation, design reviews, fielding/implementation decisions, and other acquisition-related milestone decision or knowledge points. These metrics also aid in developing criteria for contract incentive structures to motivate achievement of mission or business outcomes. For further details, see the article "Acquisition Management Metrics."

Analyses of alternatives (AoAs) help justify the need for starting, stopping, or continuing an acquisition program [6]. As such, they may occur at any point in the system (or acquisition management) life cycle. Typically they occur before initiating an acquisition program, or in the case of IT programs subject to the requirements of the Clinger-Cohen Act, they may be used to make capital planning and investment decisions based on a business case analysis. The decision criteria and associated metrics used to select the materiel (or capital investment) alternative should serve as the basis for the metrics used to plan and manage the acquisition effort. For more on AoAs, see the article "Performing Analyses of Alternatives."

Performance-based acquisition. Identifying outcome-focused metrics to manage the acquisition effort is consistent with the concept of performance-based acquisition (PBA). PBA (formerly performance-based contracting) is a technique for structuring an acquisition based on the purpose and outcome desired, instead of the process by which the work is to be performed [7]. PBA provides insight into key technical enablers for achieving the required level of performance for a product or the level of service when acquiring services. These factors and associated metrics (preferably quantitative) should serve as a subset of the key performance parameters that define success of the acquisition effort. They should be folded into the measurable criteria for an integrated master plan and integrated master schedule. Cost and schedule must also be considered since achievable technical performance or quality of service is often related. For more information, see the "Integrated Master Schedule (IMS)/ Integrated Master Plan (IMP) Application" article in this topic. For more information on costing, see the article "Life-Cycle Cost Estimation."

Using technology as an advantage. Technology can be both an enabler and an inhibitor. Acquiring new technology for a system that largely depends on IT or on emerging technology means planning, managing, and executing a program plan centered on reducing the risks associated with IT or the new technology. Use of research and development activities, prototyping, continuous technical assessments, and planned incremental deliveries are ways to mitigate the risks. The program plan should be designed to adapt to either technology evolution or user needs during the program life cycle. Transitioning new technology into the users' hands is also a risk to be considered and managed. For more on technology and transition, see the articles "Assessing Technical Maturity" and "Technology Planning."

Leveraging increased specialization. As the discipline of systems engineering evolves, it increases in span or scope into areas like system-of-systems engineering and enterprise systems engineering (see the SEG's introductory article "The Evolution of Systems Engineering"). It also leads to finer division and differentiation of systems engineering skills. As a result, more than ever, systems engineering is a team sport in which SEs must collaborate with and orchestrate the efforts of specialty SEs to achieve program success.

Although performance engineering is recognized as fundamental in manufacturing and production, its importance earlier in the life cycle is often overlooked. Performance engineering activities are most often associated with hardware and software elements of a system. But, its principles and techniques can be applied to other aspects of systems that can be measured in some meaningful way, including, for example, business processes. The article "Performance Engineering" discusses these two issues and describes where and how to employ the discipline across the systems engineering life cycle.

SEs will frequently team with economic/cost analysts at various stages of the acquisition life cycle to perform investment analyses, and their familiarity with key aspects of prevalent analytic approaches can improve the overall quality of completed analyses and the efficiency of supporting activities performed. The article "Comparison of Investment Analyses" describes the key aspects of investment analyses, including decisions supported; primary objectives; and general approaches that can be applied.

References and Resources

1. Federal Acquisition Regulation (FAR).
2. Department of Defense, December 8, 2008, Department of Defense Instruction No. 5000.2.
3. Space and Missile Systems Center U.S. Air Force, April 29, 2005, *SMC Systems Engineering Primer and Handbook.*
4. 104th U.S. Congress, "Capital Planning and Investment Control," Information Technology Management Reform Act of 1996 (now the Clinger/Cohen Act), Section 5112.
5. Office of Management and Budget (OMB), August 2009, OMB Circular A-11.
6. Office of Aerospace Studies, July 2008, *Analysis of Alternatives Handbook.*
7. U.S. General Services Administration (GSA), Performance-Based Acquisition.

Additional References and Resources

"Program Formulation and Project Planning," MITRE Project Leadership Handbook.

"3.2 Government Acquisition Support," MITRE Systems Engineering Competency Model.

Definition: *An analysis of alternatives (AoA) is an analytical comparison of the operational effectiveness, cost, and risks of proposed materiel solutions to gaps and shortfalls in operational capability. AoAs document the rationale for identifying and recommending a preferred solution or solutions to the identified shortfall(s) [1].*

Keywords: *analysis of alternatives, AoA, baseline alternative, cost analysis, criteria, evaluation, materiel solutions, risk analysis*

ACQUISITION PROGRAM PLANNING

Performing Analyses of Alternatives

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of AoA and where it occurs in the acquisition process. MITRE SEs are also expected to understand and recommend when an AoA is appropriate to a situation. They are expected to develop, recommend, lead, and conduct technical portions of an AoA, including strategies and best practices for execution. SEs are expected to monitor and evaluate AoA technical progress and recommend changes when warranted.

Background

An AoA is one of the key documents produced in preparation for major milestones or programs reviews. It is especially key at the start of a new program. Very often, MITRE's systems engineering support to programs that are in pre-Milestone A or B phases involves support to AoA efforts. Recommendations from the AoA determine the procurement approach for either a new program or the continuance of an existing program. MITRE SEs involved in program planning are frequently called on to participate in these analyses and lead the technical efforts associated with assessing other existing programs for applicability to the mission, cost-effectiveness, and risk. MITRE SEs provide necessary systems engineering skills (requirements analysis, technical evaluation, architecture, etc.), freedom from bias, and access to subject matter expertise required for AoAs.

Why Do We Do AoAs?

AoAs are performed to allow decision makers to understand choices and options for starting a new program or continuing an existing program. The bottom line is cost effectiveness through non-duplication of effort and lowest risk to successful program delivery. An example is a MITRE customer who determined there were gaps in their current suite of systems to meet anticipated needs. One of the systems was limited in capability and nearing its end of life with no follow-on program of record to continue or improve it. The customer needed an analysis of existing systems and technologies to make informed decisions on the best path forward to provide an integrated solution of systems to meet their emerging needs. The Departments of Defense (DoD), Homeland Security, and Treasury require AoAs for new procurements. The USAF (Air Force Materiel Command) has one of the more robust documented processes for performing an AoA [1]. Per DoDI 5000.02, the purpose of the AoA is to assess the potential materiel solutions to satisfy the capability need documented in an initial capabilities document [2]. Approval to enter into the development phase of a program is contingent on completion of an AoA, identification of materiel solution options by the lead DoD component that satisfy the capability need, and satisfaction of phase-specific entrance criteria for the development milestone.

Commercial industry also uses "alternative analyses," but they are usually more focused on life-cycle cost. An equally interesting and valuable feature of an AoA is an assessment of risk—including operational, technical, and programmatic risks. This kind of assessment is not widely seen in AoA processes, and is not always employed [3]. As has been seen repeatedly with other systems engineering disciplines, a good process is a good start, but it does not guarantee success unless it is actually followed.

Best Practices and Lessons Learned

The plan is important. A major step leading to a successful AoA is the creation of a well-considered study plan. The study plan establishes a roadmap for how the analysis should proceed, who is responsible for doing what, and why it is being done [1]. It should include the following information:

- Understand the technology gaps and capability gaps—what needs is the intended system supposed to meet?
- Develop viable alternatives:
  - Define the critical questions.
  - List assumptions and constraints.
  - Define criteria for viable/nonviable.
  - Identify representative solutions (systems/programs).
  - Develop operational scenarios to use for comparisons/evaluation.
- Identify, request, and evaluate data from the representative systems/programs (determined to be viable).
- Develop models—work through scenarios.

The AoA should "assess the critical technology elements associated with each proposed materiel solution, including technology maturity, integration risk, manufacturing feasibility, and, where necessary, technology maturation and demonstration needs [3]."

Sufficient resourcing is key. Allocate sufficient resources and time for completing each of the actions and assessments and for preparing the final analysis product. The biggest risk to success is the lack of time to adequately perform the AoA. Compressed schedules associated with preparing for a new procurement, and the associated execution of funding, can present major problems for the AoA team. If faced with this issue, resources may need to be increased and allocated full time to the effort.

Know the baseline before starting the AoA. For many AoAs, a capability exists but it is either nearing its end of life or no longer satisfies current needs. In these cases, it is critical to first understand the existing capability baseline. The set of alternatives considered in the AoA must include an upgrade path from the status quo. Unless information about the existing capability (referred to as the "baseline alternative") is already in hand, it is necessary to ensure that sufficient effort is planned during the AoA to capture that baseline fully enough for the comparison analysis to be performed. The point of comparison is likely to be in the future (at which time it is projected that the new capability would field), so there may be a need to project an upgrade path for the existing capability for fair comparison. SEs, design engineers, and software engineers should get involved early to review the baseline, understand it, and project an upgrade path for the "baseline alternative."

Know your stakeholders. Understand the stakeholders and decision makers involved with the AoA and how they will use the results. Assess the political, operational, economic, and technical motivations of the stakeholders to inform the scope of the analysis and its focus. Use community and user stakeholders to assist in determining the objectives, and then the measures and metrics that will be used to identify the "best" solution. Not only does this leverage their knowledge, it provides a means to obtaining their buy-in on the result.

Beware premature convergence. A recent GAO (Government Accountability Office) report on defense acquisitions attributes premature focus on a particular solution or range of solutions as a failing of AoAs [3]. If stakeholders are already enamored of a particular solution, completing a full AoA may be difficult. The intention is to survey a broad range of alternatives to ensure the best value and technical match to the need. A narrow scope or attention paid to a particular solution renders the AoA ineffective for decision making and leads to increased risk in the resulting program.

Know your AoA team. Establish standards for the minimum level of expertise and experience that AoA study participants in each integrated project team/working group must meet. Subject matter experts (SMEs) should truly be recognized experts in their area, rather than just adding a particular organizational affiliation.

Understand the mission. It takes focusing on both the mission and the technical capabilities to be able to perform an adequate assessment. AoAs should make use of simulation and modeling to help determine the best solution; but a simulation is only as good as its fidelity and the input parameters (i.e., subject to "garbage in, garbage out"). Make use of community SMEs and users to ensure a good understanding of the objective operations needed to meet the gaps. MITRE SEs need to possess both the operational knowledge and technical skills to adequately analyze technical solutions. The objective is to be credible, thorough, and comprehensive. A good AoA needs to address the full spectrum of mission, operating environment, technologies, and any other unique aspects of the program. It also needs to be articulated well enough to enable decision makers to make an informed decision.

Obtain technical descriptions of the materiel solutions. Frequently the selection of alternatives for the study is made by team members who are focused on the operational mission and capability gaps. Yet technical knowledge must also be applied by the AoA team or time and resources may be wasted investigating alternatives that are not technically feasible. Early application of technical criteria will avoid this. Ask for technical descriptions (e.g., technical description documents) as well as operational descriptions of the alternatives before starting the AoA. Doing an early DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities) analysis makes it possible for the AoA to focus its efforts primarily on dealing with feasible material solutions.

Anticipate problems. Analyzing a broad range of solutions involves collecting a considerable amount of information on the representative systems/programs as well as on other details like operating environment or communications infrastructure. Industry requests for information can be useful, but do not always produce the detail needed. Look for other data sources through government, contractor, or MITRE contacts. This issue can also be exacerbated by a compressed schedule, leading to inaccurate or incomplete analyses due to lacking detailed information. In any case, a good assumption going into an AoA is that all necessary information will not be forthcoming and will necessitate creating some workarounds (e.g., facsimile, assumptions). Be persistent!

Leverage MITRE. A wide range of technical expertise and system/program expertise is required for an AoA. Determine what skills are required for the AoA plan and leverage what you can from the broader MITRE team. Not only should MITRE expertise be used for the technical expertise in the areas of technology applicable to the solution set, but also for analysis techniques, and modeling and simulation to aid in the evaluation process. MITRE has in-house expertise in engineering cost modeling and life-cycle cost trade-off analysis that can be leveraged as well. As an example, MITRE's participation in a particular AoA involved creating and using models to assist in the evaluation as well as providing subject matter expertise in the technical areas of radio frequency, electro-optical, infrared, high-power microwave, and electronic warfare technologies. This led to a very successful AoA. Last, nothing can replace a good face-to-face discussion among novice and experienced engineers on an AoA.

Beware the compressed schedule. As mentioned above, an inadequate timeframe to conduct an AoA can render its conclusions ineffective. The GAO found that many AoAs have been conducted under compressed timeframes—six months or less—or concurrently with other key activities that are required for program initiation in order to meet a planned milestone decision or system fielding date. Consequently AoAs may not have enough time to assess a broad range of alternatives and their risks, or may be completed too late in the process to inform effective trade discussions prior to beginning development [1].

Incorporate the risk analysis. Risks are important to assess because technical, programmatic, or operational uncertainties may be associated with different alternatives that should be considered in determining the best approach [1].

The GAO reported that some of the AoAs they reviewed did not examine risks at all; they focused only on the operational effectiveness and costs of alternatives. Other AoAs had relatively limited risk assessments. For example, several AoAs did not include integration risks even though the potential solution set involved modified commercial systems that would require integration of subsystems or equipment. Based on a recent Defense Science Board report on buying commercially based defense systems, programs that do not assess the systems engineering and programmatic risks of alternatives do not understand the true costs associated with militarizing commercial platforms or integrating various commercial components [3]. Other AoAs did not examine the schedule risks of the various alternatives, despite accelerated schedules and fielding dates for the programs. They also found that programs with AoAs that conducted a more comprehensive assessment of risks tended to have better cost and schedule outcomes than those that did not [1]. For more information on risk identification and management, see the Risk Management topic and articles within this section of the SEG.

References and Resources

1. Air Force Materiel Command Office of Aerospace Studies, July 2008, *Analysis of Alternatives Handbook: A Practical Guide to Analyses of Alternatives.*
2. Department of Defense Instruction 5000.02, December 8, 2008 (revised), Operation of the Defense Acquisition System.
3. *Many Analyses of Alternatives Have Not Provided a Robust Assessment of Weapon System Options,* September 2009, GAO Report.

Definition: *According to BusinessDictionary.com, metrics are defined as standards of measurement by which efficiency, performance, progress, or quality of a plan, process, or product can be assessed. Acquisition management metrics are specifically tailored to monitor the success of government acquisition programs.*

Keywords: *acquisition metrics, leading indicators, program success*

ACQUISITION PROGRAM PLANNING

Acquisition Management Metrics

MITRE SE Roles and Expectations: Within the role of providing acquisition support, MITRE systems engineers (SEs) are tasked with understanding technical risk and assessing success. Management metrics are used as a mechanism to report progress and risks to management. MITRE staff should understand how these metrics influence and relate to acquisition systems engineering.

The use of metrics to summarize a program's current health, identify potential areas of concern, and ability to be successful are common practice among government departments, agencies, and industry. Metrics range from detailed software metrics to more overarching program-level metrics. Some of the following examples are derived primarily from the

Department of Defense (DoD) program practice, but the principles are applicable to any program.

Probability of Program Success Metrics

As an aid in determining the ability of a program to succeed in delivering systems or capabilities, the military services developed the Probability of Program Success (PoPS) approach. PoPS standardizes the reporting of certain program factors and areas of risk. Each service measures a slightly different set of factors, but all the tools use a similar hierarchy of five factors at the top level. These factors are Requirements, Resources, Execution, Fit in Vision, and Advocacy. Associated with each factor are metrics, as indicated in Figure 1. These tools are scoring methodologies where metric criteria are assessed by the program office, a metric score/point is determined, and metric scores are weighted and then summed for an overall program score. The summary score is associated with a color (green, yellow, or red), which is the primary way of communicating the result. It is at the metric level and the criteria used to assess the metric where the biggest differences between the tools exist. Each tool weighs metrics differently, with Air Force and Navy varying these weights by acquisition phase. Furthermore, each service uses different criteria to assess the same or similar metric. See the references [1, 2, 3, 4] for each service tool to better understand how metrics are scored.

Best Practices and Lessons Learned: Determining the value of each metric is the responsibility of the acquisition program team. Systems engineering inputs are relevant to most of the reporting items; some are more obvious than others. With respect to staffing/resources, it is important to understand the right levels of engineering staffing for the program management office and the prospective development contractor to ensure success. At the outset of an acquisition, a risk management process should be in place (see the SEG's Risk Management section); the ability of this process to adequately identify and track risks is a major component of the PoPS tool. All technical risks should be incorporated in this assessment, including those that may be included in the technical maturity assessment. Immature technology can be a considerable risk to program success if not managed appropriately; it also can be scheduled for insertion into the program delivery schedule on maturation. For more detail on technology maturity, see the article Assessing Technical Maturity.

Note: This structure is generic and meant to closely represent what the services capture in their respective PoPS tools and where.

Although a metric name may be different or absent when comparing one tool to another, same or similar qualities may be captured in a different metric. Conversely, metrics may have the same or similar name but capture different qualities of the program. See the individual service's PoPS operations guide for details [1, 2, 3, 4].

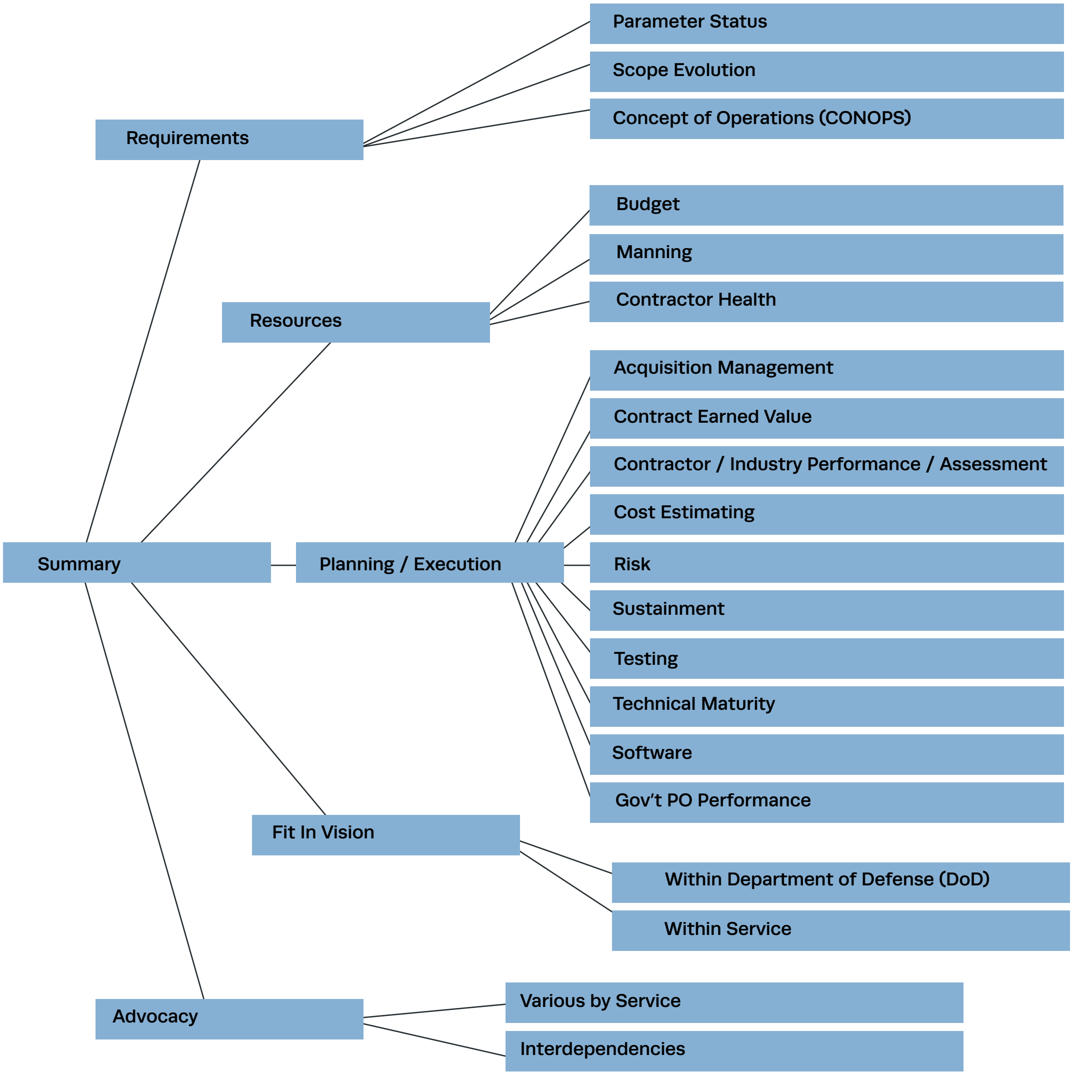


Figure 1. Generic Representation of Metrics Considered in PoPS Tools

Earned Value Management (EVM) Metrics

A subset of program management metrics is specific to contractor earned value. Typical EVM metrics are the Cost Performance Index and the Schedule Performance Index; both are included in each of the service's PoPS tool [5, 6]. Although EVM is mostly considered a monitoring tool for measuring project performance and progress, it is also a planning tool. Using EVM effectively requires the ability to define, schedule, and budget the entire body of work from the ground up. This is something to be considered in the planning phase of an acquisition program (see the article "Integrated Master Schedule (IMS)/Integrated Master Plan (IMP) Application") because it is closely linked to the Work Breakdown Structure (WBS).

Best Practices and Lessons Learned: Fundamental to earned value is linking cost and schedule to work performed. However, work performed is often specified at too high a level to identify problems early. This is linked back to the generation of the WBS during the initial program planning and whether it was created at a detailed enough level (i.e., measureable 60-day efforts) to clearly define work performed or product developed. In cases where the detail is insufficient, EVM is unlikely to report real problems for several months. It is usually program engineers and acquisition analysts who are able to identify and report technical and schedule problems before the EVM can report them. Another method is the use of Technical Performance Measures (TPMs). TPMs are metrics that track key attributes of the design to monitor progress toward meeting requirements [7, 8]. More detailed tracking of technical performance by contractors is becoming popular as a way to measure progress and surface problems early using Technical Performance Indices at the lowest product configuration item (i.e., Configuration Item, Computer Software Configuration Item) [9].

Appropriate insight into evaluating the work performed for EVM can be challenging. It often requires close engineering team participation to judge whether the EVM is accurately reporting progress. A case where this is particularly challenging is in large programs requiring cross-functional teams and subcontracts or associate contracts. Keeping the EVM reporting accurate and timely is the issue. To do this, the contractor's team must have close coordination and communication. Check that forums and methods are in place to accurately report EVM data for the entire program.

Systems Engineering Specific Metrics—Leading Indicators

Several years ago, MITRE engineers assisted in the development of a suite of "leading indicators" to track more detailed systems engineering activities for a DoD customer. Table 1 summarizes some of these metrics (expanded to generically apply here), which were to be assessed as green, yellow, or red, according to specified definitions. An analysis of the overlap of the leading indicators with PoPS metrics was conducted. Although several metrics have similar names, the essence of what is captured is different, and there is very little overlap.

Table 1. Leading Indicators

Leading Indicator Area	Detailed Metrics	Measurement Comments
Program Resources	Required Staffing Applied Appropriate Skills Applied Churn Rate Training Available	Staffed to at least 80% according to plan Ideally churn is less than 5% per quarter
Requirements	Volatility/Impact	Low volatility (< 5%) can still be a problem if the changing requirements have a large impact to the cost and schedule (like 10–15% cost growth)
Risk Handling	Trend Appropriate Priority Applied	Are the risks coming to closure without significant impact? Alternatively, are there more risks being created over time (+ or – slope)? Are resources being applied to the risk? Is there appropriate engineering and PM oversight?
Interoperability	Community of Interest Established and Active Data Sharing Addressed Network Accessibility	Ideally, a data sharing plan exists for members of the COI and addresses the data formats, visibility / discovery (to include metadata), and plan for exposure.
Software Development	Sizing Quality Productivity Defects	(These are fairly standard metrics—for more information on software metrics, see NIST Special Publication 500-234.)
Verification and Validation	Complete Requirements Documentation Requirements Test Verification Matrix (RTVM) Verification and Validation (V&V) Plan	The degree of completeness and volatility of these items is the measurement.
Technology Readiness	Technology Readiness	TRL of at least 6
Risk Exposure	Cost Exposure	15% deviation indicates high risk.
	Schedule Exposure	
Watchlist Items (Risks)	Severity Closure Rate	As opposed to the risk handling metric above, this one looks at the severity of the risks—which ones are likely to occur with a med-high impact to the program.

Although this suite of "leading indicators" was not officially implemented by the customer, they make a good set of items to consider for metrics; they also capture areas not covered in other models.

Best Practices and Lessons Learned

Be careful comparing metrics across different tools. Although the metric names in the three cited tools (and others) may be similar, they may be assessed differently. When comparing metrics across different tools, you need to understand the details of metric definitions and assessment scales. Make sure this is conveyed when reporting so that the intended audience gets the right message; the assessment needs to stand on its own and not be misinterpreted.

Understand what a metric is supposed to be measuring. For example, trends, current status, and the ability to be successful when resolution plans are in place. This will ensure that results are interpreted and used properly.

Use the metrics that are most appropriate for the phase of the program you are in. If the program has overlapping phases, use multiple metrics. When a program in overlapping phases is assessed as if it were in a single program phase (as in PoPS), the resulting report is usually not an accurate representation of the program status.

Be cautious of methodologies where subjective assessments are converted to scores. Developing scoring methodologies can appear to be simple, yet mathematical fundamentals must still be followed for results to be meaningful [10]. This is particularly a concern when the resulting single score is taken out of context of the analysis and used as a metric in decision making.

References and Resources

1. U.S. Army, May 2004, *Probability of Program Success Operations Guide.*
2. U.S. Air Force, July 2008, *Probability of Program Success (PoPS) Model, SMART Integration, Operations Guide,* Ver. 1.0.
3. Department of the Navy, September 2008, *Naval PoPS Guidebook, Guidance for the Implementation of Naval PoPS, A Program Health Assessment Methodology for Navy and Marine Corps Acquisition Programs*, Ver. 1.0.
4. Department of the Navy, September 2008, *Naval PoPS Criteria Handbook, Supplement for the Implementation of Naval PoPS*, Ver. 1.0.
5. Meyers, B., Introduction to Earned Value Management, EVM 101, ESC/AE (Acquisition Center of Excellence).
6. Meyers, B., Analysis of Earned Value Data, EVM 401, ESC/AE (Acquisition Center of Excellence).

7. Ferraro, M., “Technical Performance Measurement,” Defense Contract Management Agency, Integrated Program Management Conference, November 14–17, 2001.
8. Pisano, Commander N. D., *Technical Performance Measurement, Earned Value, and Risk Management: An Integrated Diagnostic Tool for Program Management*, Program Executive Office for Air ASW, Assault, and Special Mission Programs (PEO [A]).
9. “Statement of Work (SOW) for Development, Production, Deployment, and Interim Contractor Support of the Minuteman Minimum Essential Emergency Communications Network (MEECN) Program,” June 28, 2007, MMP Upgrade.
10. Pariseau, R., and I. Oswalt, Spring 1994, “Using Data Types and Scales for Analysis and Decision Making,” *Acquisition Review Quarterly,* p. 145.

Additional References and Resources

Massachusetts Institute of Technology, International Council on Systems Engineering, Practice Software and Systems Measurement, June 15, 2007, *Systems Engineering Leading Indicators Guide*, Ver. 1.0.

NIST Special Publication 500-234, March 29, 1996, Reference Information for the Software Verification and Validation Process, (Appendix A.1, Metrics).

Definition: *Assessing the maturity of a particular technology involves determining its readiness for operations across a spectrum of environments with a final objective of transitioning it to the user. Application to an acquisition program also includes determining the fitness of a particular technology to meet the customer's requirements and desired outcome for operations.*

Keywords: *disruptive technology, emerging technology, mature technology, revolutionary technology, sustaining technologies, technological innovation, technology assessment, technology insertion, technology readiness, TRL*

ACQUISITION PROGRAM PLANNING

Assessing Technical Maturity

MITRE SE Roles and Expectations: Systems engineers (SEs) are expected to anticipate future technology needs and changes based on a broad understanding of the systems context and environment, recommend long-term technology strategies that achieve business/mission objectives, and exploit innovation. As part of acquisition planning, the ability to successfully procure new technology and systems involves assessing current technology to support the program requirements. Understanding how to assess technology readiness, apply technologies to a program, and mature technologies for insertion is an important part of what MITRE SEs are expected to provide our customers. And because MITRE serves a role independent from commercial industry, MITRE is often asked to independently assess a particular technology for "readiness."

Whether assessing the usefulness of a particular technology or research program, or assessing the ability to meet a set of new requirements with mature technology, it is best to first understand the typical cycle technology developments follow and the methodologies to consider for selecting the appropriate path for your program.

Best Practices and Lessons Learned

Technology hype cycle. One way to look at technology maturity is through a Gartner hype cycle [1]: a graphic representation of the maturity, adoption, and business application of specific technologies. Gartner uses hype cycles to characterize the over-enthusiasm or "hype" and subsequent disappointment that typically follow the introduction of new technologies. A generic example of Gartner hype cycles is shown in Hype Cycles (Figure 1).

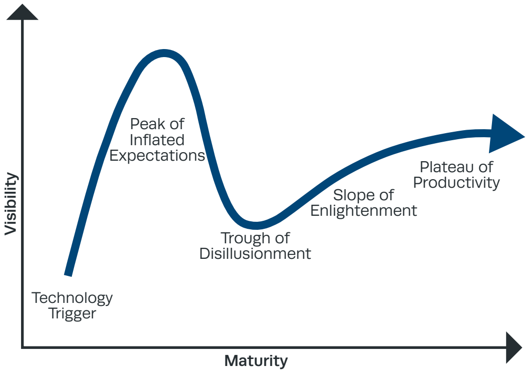


Figure 1. Hype Cycles

A hype cycle in Gartner's interpretation has five steps:

1. **Technology Trigger:** The first phase of a hype cycle is the "technology trigger" or breakthrough, product launch, or other event that generates significant press and interest.
2. **Peak of Inflated Expectations:** In the next phase, a frenzy of publicity typically generates over-enthusiasm and unrealistic expectations. There may be some successful applications of a technology, but there are typically more failures.
3. **Trough of Disillusionment:** Technologies enter the "trough of disillusionment" because they fail to meet expectations and quickly become unfashionable. Consequently, the press usually abandons the topic and the technology.
4. **Slope of Enlightenment:** Although the press may have stopped covering the technology, some businesses continue through the "slope of enlightenment" and experiment to understand the benefits and practical application of the technology.
5. **Plateau of Productivity:** Mainstream adoption starts to take off. Criteria for assessing provider viability are more clearly defined. The technology's broad market applicability and relevance are clearly paying off.

Although Gartner references the "press" above, technology hype can and does occur throughout different organizations. It can often result in significant program investment funding being applied to technologies that may not be suitable for the intended system or user, but were deemed promising by program stakeholders. The preceding steps are applicable to all MITRE sponsors and customers to which technology programs are marketed. When significant attention is given by program stakeholders to a new research, technology, technology development program, or demonstration, the targeted technology should be objectively evaluated and assessed for maturity as soon as possible before committing any significant program investment funding.

Technology maturity. A generic depiction of technology maturity is shown by the s-curve in Figure 2. In general, technology can be defined as follows: new technology has not reached the first tipping point in the s-curve of technology maturity; improving or emerging technology is within the exponential development stage of the curve after the first tipping point and before the second tipping point; mature technology follows the second tipping point before the curve starts down, and aging technology is on the downward tail.

The most universally accepted methodology for assessing the upward slope of this curve is the Technology Readiness Level (TRL) scale [2]. There are actually several versions of the original NASA-developed TRL scale depending on the application (software, manufacturing, etc.), but all rate a technology based on the amount of development completed, prototyping, and testing within a range of environments from lab (or "breadboard") to operationally relevant. It is critical to get a common and detailed understanding of the TRL scale among program stakeholders, particularly concerning terms like "simulated environment," "relevant environment," and "operational mission conditions," which must be interpreted in the context of the system or capability under development. Close communication among the program office, operational users, and the developer on these terms is needed to ensure an accurate assessment. One factor the current TRL scale does not address is how well the developed technology fits into the architecture and system structure of the program absorbing it. This is an integral part of the systems engineering job and critical to the success of the technology transition.

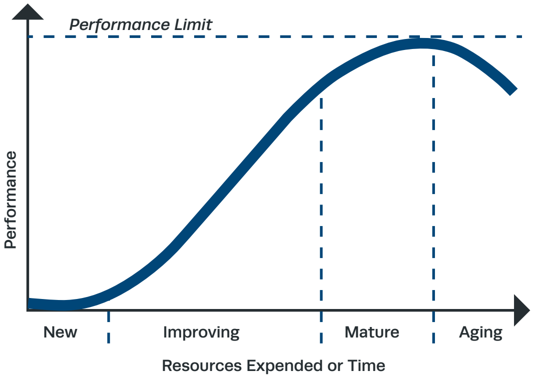


Figure 2. Technology Maturity

Selecting technology alternatives. For assessing which technology to employ to satisfy new requirements, various fitness criteria can be used to select which alternative will best realize customer desired outcomes from the total spectrum of technologies available. Criteria that consider both the technology and the customer's ability to assimilate it are more likely to succeed than those that consider only the technology (as seen above in the use of TRLs). Moore [3] identifies types of customers as: innovators, early adopters, early

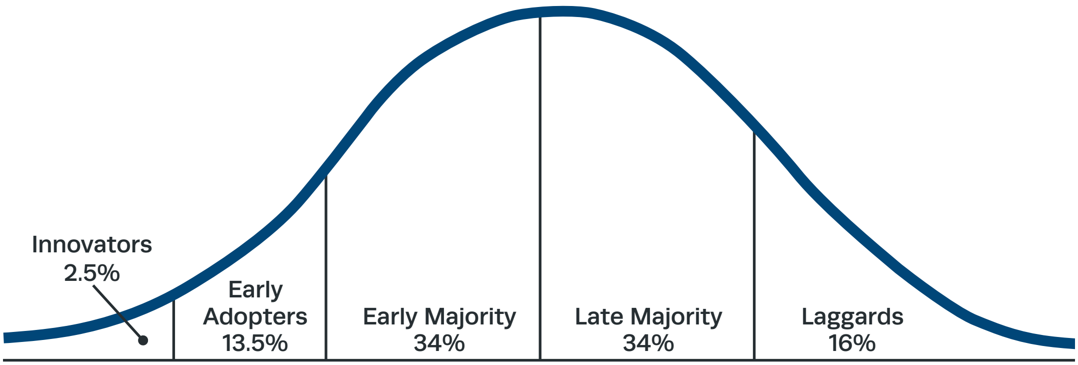


Figure 3. Roger's Bell Curve

majority, late majority, and laggards. The curve depicted in Figure 3 is referred to as the technology adoption life cycle, or Roger's Bell Curve [4].

As the names suggest, each customer type has its own tolerance for change and novelty. Technology assessment considers the customer's tolerance for disruptive change as well as new or old technologies. For example, it would not be appropriate to recommend new technology to "late majority" customers, nor mature technology to "innovators."

Department of Defense acquisition programs are required to assess all threshold capabilities in the Capabilities Description Document for maturity; those deemed to be met with immature technology (a TRL of less than six) will not be considered further as "threshold" and may jeopardize the program milestone decision. Programs structured to inject developing technologies could be more receptive to innovation and less mature technologies, but in this case be sure to carefully evaluate the risks involved (for further reference, see the SEG's Risk Management topic).

ABC alternatives. Another dimension of the selection criteria considers the capabilities of technology providers. Former Director of the Defense Information Systems Agency, Lt Gen Charles Croom, devised a new philosophy for acquisition called ABC [5]. In the "ABC" concept, "A" stands for adopt existing technology, "B" is buy it, and "C" is create it yourself. Adopt may seem an obvious decision if the technology fits the purpose, but both the technology and the provider should be evaluated for reliability and sustainability. With the buy alternative, vendor responsiveness and capability are concerns (for further reference, see the SEG's Integrated Logistics Support topic). Create is the choice of last resort, but it may be the best alternative in certain circumstances.

References and Resources

1. Gartner Hype Cycles, http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp, accessed February 4, 2014.
2. “Technology Readiness Level,” Wikipedia, accessed February 22, 2010.
3. Moore, G. A., 1998. *Crossing the Chasm*, Capstone Publishing Limited.
4. Rogers, E., Diffusion of Innovation Model, “Technology Adoption Lifecycle,” Wikipedia, accessed February 22, 2010.
5. Gallagher, S., March 20, 2008, “Croom: Acquisition Done Better, Faster, Cheaper,” *Federal Computer Week*.

Definition: *Technology Planning is the process of planning the technical evolution of a program or system to achieve its future vision or end-state. Technology planning may include desired customer outcomes, technology forecasting and schedule projections, technology maturation requirements and planning, and technology insertion points. The goal is a defined technical end-state enabled by technology insertion over time. Note that sometimes this is referred to as "strategic technical planning" (STP) applied at a program level, although the preferred use of the STP term is at the enterprise or portfolio level [1].*

Keywords: *technology evaluation, technology plan, technology planning, technology roadmap*

ACQUISITION PROGRAM PLANNING

Technology Planning

MITRE SE Roles and Expectations: MITRE's role as a strategic partner with our customers requires us to focus on the up-front planning stages of customer programs, including defining the technical direction of a particular program. MITRE systems engineers (SEs) working on technical strategy and planning are expected to understand the future vision and mission being addressed by that plan and how technology can be brought to bear on solutions to meet that future vision. MITRE SEs are also expected to acquire and maintain insight into developing technology to provide a timely "honest broker" perspective in technology planning. MITRE SEs are also expected to bridge user and research communities to better align government research investment and direction with future operational mission needs.

What Is Technology Planning?

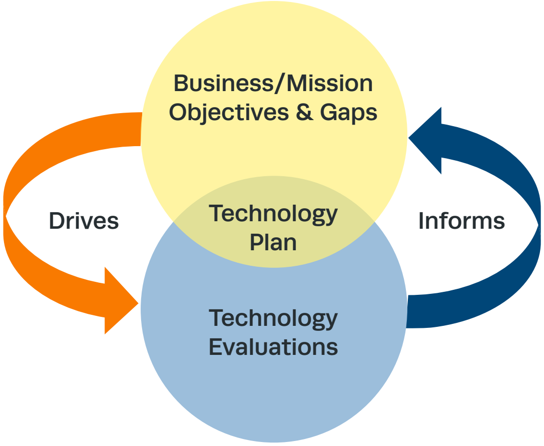


Figure 1. Technology Planning Bridges Business, Mission, and Technology Domains

For a particular acquisition program, the future technical direction may be defined by generating a program-level technical strategy documented in a technology plan or roadmap. Given the current state and constant change of technology, without common guidance, individual organizations may use their own methods and technologies in ways that can actually hinder adaptation to the future. A technology plan provides the guidance to evolve and mature relevant technologies to address future mission needs, communicate vital information to stakeholders, provide the technical portion of the overall program plan (cost and schedule), and gain strong executive support. It should be a "living" document that is a basis for an ongoing technology dialog between the customer and the systems developers.

Strategic technical planning embraces a wider scope and can cover a wide range of topics. It can be organizationally dependent, portfolio focused, enterprise-wide, and system focused. A program's technology plan may be linked to an organizational or enterprise "strategic technical plan" [2, 3]. It should also serve as the companion to the program's business or mission objectives because business or mission needs and gaps drive the technology needs. At the same time, technology evaluations inform the technical planning activity of technologies to achieve the future technical vision or end-state. The resulting technology plan serves as the roadmap for satisfying the gaps over time to achieve the end-state. These relationships are depicted in Figure 1.

Technology Plan Components

A technology plan is a key enabler for the systems engineering function. Based on the future mission or business needs, it defines a desired technical end-state to evolve toward. Because that end-state may not be achievable with current technology, it is important to determine which technologies are available now, which technologies are in development, including their maturity levels, and which technologies do not yet exist. This helps influence an investment strategy that can focus and push the "state of the art," and it helps define requirements that are not achievable at all or may be cost prohibitive.

Technologies requiring further investment and maturation should be assessed as part of the technical planning process. Appropriate risk should be assigned to technologies assessed as immature, with the need for concomitant mitigation plans. Technologies that have been in the research and development (R&D) phase for an extended period (over five years) should be assessed for the maturation trend to determine if additional investment would significantly improve the maturity.

At a minimum, the plan should include identification of all technology being brought to bear for the solutions, the maturation and trend of applicable technologies (forecast), insertion points, required investments, and dependencies.

Best Practices and Lessons Learned

The process of developing and implementing a technology plan should include the following activities [4]:

Evaluate the environment for innovative uses of technology. What is changing in the environment that needs to be taken into account or can be exploited? Where is industry headed and what are its technology roadmaps?

Define desired results. Where does the organization want to be within a planning horizon, usually 5–10 years? Envision the future as if it were today, and then work back to the present.

Identify the core technologies needed for meeting the vision and focus on those first. Assess the risks for maturation and focus on investment and mitigation. If the risk is very high, the choice is to wait and depend on the "bleeding edge," or embark on a serious investment program. The criticality of the technology and/or mission will drive this choice. If it is indeed a core technology and critical to the success of achieving the end-state, significant investment will need to be applied to buy down the risk. One example of this is the government choosing to invest heavily in cyber security.

Identify the remaining technologies applicable to the mission or business area end-state. But, don't become enamored with technology for technology's sake! Keep it simple and focused on the end-state.

Establish a quantifiable feedback system to measure progress. Define what must be done and how it will be measured to determine progress. Define measures of success to gauge whether the implementation of the plan is progressing successfully. Adjust the plan accordingly. Measuring return on investment for those technologies requiring maturation can be challenging; make allowances for failures depending on the assessed risk.

Assess the current state of the organization implementing the plan. Are resources (staff, funding) and processes in place to effectively implement the plan? Are the required skills available?

Develop tactical plans with measureable goals to implement the strategy.

Form the roadmap. Develop the phasing, insertion points, associated R&D investments, work plans or packages, and sequence the activities within each functional and major program area in the tactical plan to form the roadmap. Allocate resources and tasks and set priorities for action during the current year.

Assess the life-cycle costs of technology. Try not to underestimate the life-cycle cost of technology. This can be difficult. Industry investments in new technology tend to be closely held and proprietary. Often, the full product or prototype is not made visible to the customer until it's ready for sale or deployment. So, usually there is a lack of technical detail and understanding of the whole product or technology and its application to the mission area. The result can be increased costs for integration, maintenance, and licensing. Licensing for proprietary special-purpose (non-commercial-off-the-shelf) technology can be particularly costly. An alternative is a sole-source relationship.

Educate the organization and stakeholders on the plan and its implementation. Communicate with stakeholders and users using their operational terminology and non-technical language. Users won't support what they can't understand and can't clearly link to their mission. Communicate the plan to outside industry, labs, and associated R&D activities to ensure understanding and form or solidify relationships. The technology plan can be a tool to collaborate with industry, labs, and other organizations on shared investment strategies toward achieving common goals.

Implement the technology plan. Monitor, track, and make adjustments to the plan according to periodic reviews.

Review the technology plan. Review annually or in other agreed period by iterating the above process.

References and Resources

1. Swarz, R., and J. DeRosa, 2006, A Framework for Enterprise Systems Engineering Processes, The MITRE Corporation.
2. Byrne, R., June 2, 2005, A Netcentric Strategic Technical Plan (STP), The MITRE Corporation.
3. MITRE Mission Planning Team, November 2006, JMPS Strategic Technical Plan, Ver. 3.0.
4. CC2SG Technology Planning Team, October 5, 2005, COCOM C2 Systems Group Technical Planning Process.

Definition: *Cost analysis is "the process of collecting and analyzing historical data and applying quantitative models, techniques, tools, and databases to predict the future cost of an item, product, program or task." Cost estimates "translate system/functional requirements associated with programs, projects, proposals, or processes into budget requirements, and determine and communicate a realistic view of the likely cost outcome, which can form the basis of the plan for executing the work."[1]*

Keywords: *budget, cost analysis, cost benefit, cost estimation, regression analysis, trade-offs*

ACQUISITION PROGRAM PLANNING

Life-Cycle Cost Estimation

MITRE SE Roles and Expectations: Systems engineers (SEs) are expected to use cost analysis to identify and quantify risks and to evaluate competing systems/initiatives, proposals, and trade-offs. They are expected to collaborate with the cost/benefit analyst and sponsor/customer to define the approach, scope, products, key parameters, and trade-offs of the analysis. SEs support and provide direction to the analyst, review results, guide and evaluate the sensitivity of the analysis, and provide technical, programmatic, and enterprise-wide perspectives and context for the analyst.

Cost analysis is an often misunderstood and frequently overlooked practice that encompasses many areas of a program's business management. It combines the knowledge of many different disciplines and produces results that have far-reaching impacts on a program and its success. In many cases, the analyst who built a program's life-cycle cost estimate (LCCE) will have more knowledge and understanding of the program than any other member of the program team.

Cost Estimate Development Overview

Cost estimation methodologies and techniques vary widely depending on the customer and program. These variations are based on several factors. What is being estimated, the extent of available data, existence of an agreed-on work breakdown structure (WBS), regulatory requirements, agency requirements, and industry best practices all influence the methodologies and techniques that may be applied when creating a cost estimate. For example, the LCCE for a Department of Defense weapon system will be conducted differently and look very different from an estimate for a data center or the development of a computer application for a civilian agency. Also, the *type* of estimate will influence the methodology and approach used. A much more rigorous process is required for a budgetary estimate or a full LCCE than for a rough order of magnitude or "back of the envelope" type of estimate. Although there is no "cookie-cutter" approach to developing a cost estimate, Figure 1 depicts a generic cost estimating process.

- **Define Cost Estimate Scope:** The initial step is to define the possible scope of the cost model. The scope will determine the content of the cost elements that must be included in the model. Sources for scope definition of a program include the project management plan, the scope statement, the WBS, and any requirements documentation, etc.
- **Identify Assumptions and Constraints:** Assumptions are statements that are used to limit the scope of the model. They are a "given" as opposed to a "fact." They usually relate to a future occurrence and therefore contain uncertainty. Assumptions must be evaluated during sensitivity analysis. Constraints are usually fixed, externally imposed boundaries such as schedule, policies, and physical limitations.
- **Develop Cost Element Structure:** The cost element structure can also be thought of as a chart of accounts. It is a listing of the possible categories of cost contained in the model. Each element must be defined so that all costs are covered, and there are no duplications of costs within the structure.
- **Collect and Normalize Data:** Cost data is collected for all of the elements within the model. Information from benchmark research and actual cost experience is used. Normalization is the process for ensuring that cost data are comparable.
- **Develop Cost Estimating Relationships:** The cost data is used to develop equations that will be entered into the cost model. The equations will be the basis for estimating costs as a function of system capacity and service level.

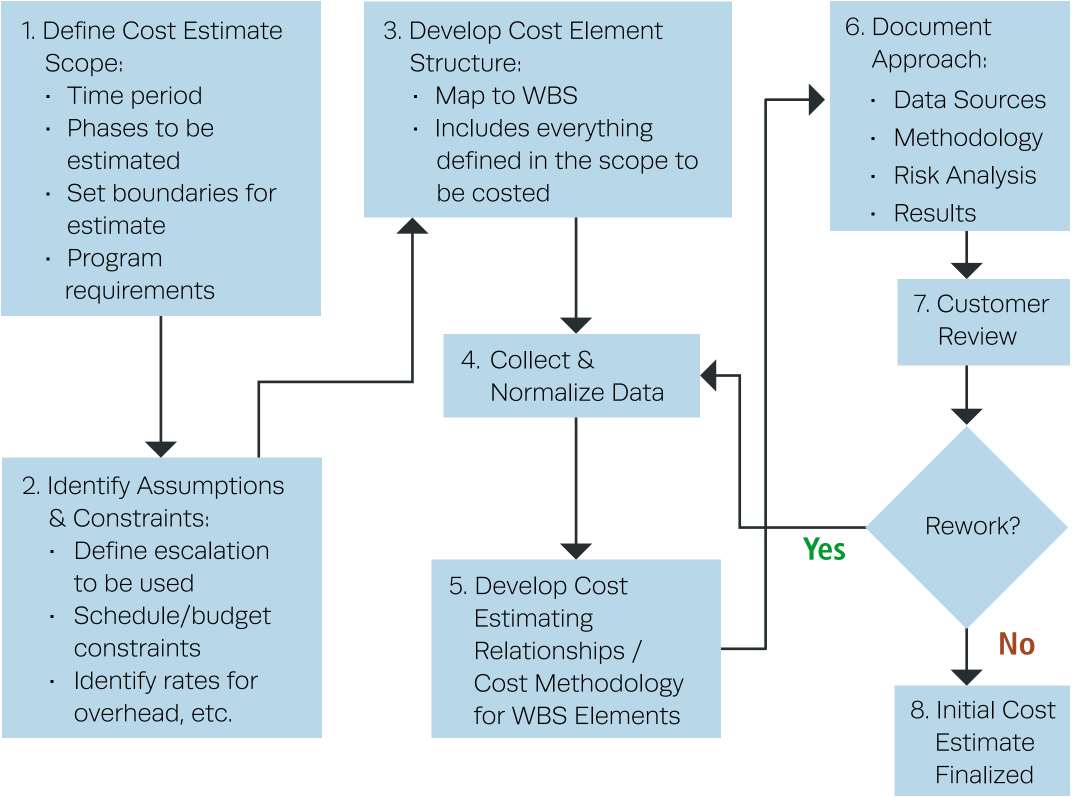

1. Define Cost Estimate Scope:
· Time period
· Phases to be estimated
· Set boundaries for estimate
· Program requirements
2. Identify Assumptions & Constraints:
· Define escalation to be used
· Schedule/budget constraints
· Identify rates for overhead, etc.
3. Develop Cost Element Structure:
· Map to WBS
· Includes everything defined in the scope to be costed
4. Collect & Normalize Data
5. Develop Cost Estimating Relationships / Cost Methodology for WBS Elements
6. Document Approach:
· Data Sources
· Methodology
· Risk Analysis
· Results
7. Customer Review
Rework?
Yes
No
8. Initial Cost Estimate Finalized

Figure 1. General Depiction of Cost Estimating Process

- **Document Approach:** Documentation is provided for each cost element that indicates the sources of data, assumptions used, and any equations that have been used in the calculations.
- **Customer Review:** A walkthrough on the model and results is conducted with the sponsor to ensure that all expected costs have been adequately represented and to achieve acceptance of the estimate.

Best Practices and Lessons Learned

The three Rs. For an LCCE to be credible and effective, it must meet three basic requirements, also known as the three Rs of cost estimation: Replication, Rationale, and Risk.

- **Replication:** The estimator must provide an audit trail that is sufficiently detailed, including clearly stated assumptions for each cost element, to allow for an independent replication of the estimate by a third or external party.
- **Rationale:** The estimator must provide a convincing and justifiable rationale for the selection of key parameter values, labor estimates, cost factors, assumptions, and all underlying inputs to the estimate. These can come from early project experience, other similar projects, parametric models, and documented engineering judgments.
- **Risk:** The estimator must conduct risk/sensitivity analysis to assess the impact of the inherent uncertainty in input values. Regression analysis is the most frequently used method of conducting sensitivity analysis in this area.

Utility to the program. Investments require clear identification of benefits, which can be categorized as either tangible or intangible. The benefits can be packaged in a program that will most likely yield desirable outcomes. As a cautionary step, one must consider the cost to stand up the program, the cost to incur the chain of activities for the identified investment such as implementation, operation, and maintenance from both quantitative and qualitative perspectives. When properly done, cost analysis provides the following utility to the program:

- Supports budgeting process by:
  - Integrating the requirements and budgeting processes.
  - Assessing affordability and reasonableness of program budgets.
  - Providing basis for defending budgets to oversight organizations.
  - Quickly/accurately determining impacts of budget cuts on program baselines and associated functionality.
- Enables early identification of potential pitfalls such as cost growth and schedule slips.
- Enables identification of future cost improvement initiatives.
- Provides for the identification and objective quantification of the impact of program risks (technical and schedule risks).
- Provides a basis for evaluating competing systems/initiatives (cost/benefit analyses and analysis of alternatives [AoA]).
- Enables proposal pricing and evaluation of proposals for cost reasonableness (independent government cost estimates).
- Captures cost impacts of design decisions to facilitate trade-offs in cost as an independent variable/design to cost/target costing.
- Facilitates evaluation of the impact of new ways of doing business (e.g., in-sourcing vs. outsourcing, commercial off-the-shelf vs. custom software).

An art, not a science. As with any discipline, the actual application and practice of cost analysis is more difficult than the academic description. It is seldom the case that the process outlined above can be applied with complete precision. In most cases many factors conspire to force the systems engineer and the cost estimator to step "outside the box" in completing a cost estimate.

When data is unavailable. Oftentimes data to support the estimate is not readily available through the customer organization. Finding supportable data will often require creative thinking and problem solving on the part of the systems engineer and cost estimator. An example is an AoA in which one of the alternatives was to build roads. The agency in question did not possess any in-house knowledge on road construction, ancillary costs (such as drainage ditches and easements), or permit and legal requirements for construction of several hundred miles of access road. The situation required reaching out to the civil engineering community and several state departments of transportation in order to bridge the knowledge gap and obtain the information in question. This resulted in a detailed and supportable estimate that the customer was able to use in justifying managerial decisions.

Adaptability is key. As stated in the cost estimation development discussion of this article, there is no single way to construct a cost estimate—too much depends on the details of the circumstances at hand. An estimator cannot do a parametric estimate, for example, if the data and situation do not support that approach. Another AoA provides an example of this. When tasked to provide an AoA for an outsourcing or internal development of a customer's financial management system, the estimator predetermined that an engineering build-up based on engineering knowledge of the problem set would be performed. Unfortunately the customer organization had no internal engineering expertise in this area. The estimator was forced to change its approach and build an estimate based on analogy of similar systems and industry benchmark studies.

Keep program needs in sight. Overall, the most important perspective on cost estimating is to keep the process in context. Remember that the cost estimate is not an end in itself but a means to an end. Understand what the program needs to accomplish through cost estimation and work with the cost estimator to tailor your product accordingly.

References and Resources

1. The International Society of Parametric Analysts and The Society of Cost Estimating and Analysis (ISPA/SCEA) Professional Development and Training Workshop proceedings, June 2–5, 2009, training presentation on Cost Estimating Basics [slide 5].

Additional References and Resources

Army Financial Management Home Page, http://www.asafm.army.mil/, accessed February 4, 2014.

GAO, March 2009, *GAO Cost Estimating and Assessment Guide*, "Best Practices for Developing and Managing Capital Program Costs."

Software Engineering Institute, http://www.sei.cmu.edu/, accessed February 4, 2014.

The Data and Analysis Center for Software, https://www.thecsiac.com/, accessed February 4, 2014.

The Project Management Institute, http://www.pmi.org/, accessed February 4, 2014.

The Society of Cost Estimating and Analysis, https://www.iceaaonline.org/, accessed February 4, 2014.

ACQUISITION PROGRAM PLANNING

Integrated Master Schedule (IMS)/Integrated Master Plan (IMP) Application

Definition: *The IMP is comprised of a hierarchy of program events, in which each event is supported by specific accomplishments, and each accomplishment is based on satisfying specific criteria to be considered complete. The IMS is an integrated, networked schedule containing all the detailed discrete work packages and planning packages (or lower level tasks of activities) necessary to support the events, accomplishments, and criteria of the IMP.*

Keywords: *earned value management, EVMS, integrated master plan, integrated master schedule, program plan, work breakdown structure, WBS*

MITRE SE Roles and Expectations: The IMS and IMP form a critical part of effectively providing acquisition support. MITRE systems engineers (SEs) should understand the use and implementation of these tools and how they can be used to effectively monitor program execution.

What We Know About the IMS and IMP

Program planning involves developing and maintaining plans for all program processes, including those required for effective program office-contractor interaction. Once the contract is signed and schedule, costs, and resources from the contractor are established, the program plan takes into account, at an appropriate level of detail, the contractor's estimations for the program. Together, the IMP and IMS should clearly demonstrate that the program is structured and executable within schedule and cost constraints with an acceptable level of risk. During the proposal evaluation and source selection phases, the IMP and IMS are critical components of the offeror's proposal; they identify the offeror's ability to partition a program into tasks and phases that can be successfully executed to deliver the proposed capability. After contract award, the contractor and/or the government use the IMP and IMS as the day-to-day tools for executing the program and tracking program technical and schedule status, including all significant risk mitigation efforts.

The IMP and IMS are business tools to manage and provide oversight of acquisition, modification, and sustainment programs. They provide a systematic approach to program planning, scheduling, and execution. They are equally applicable to competitive and sole source procurements with industry, as well as to government-only, in-house efforts. They help develop and support program/project budgeting, and can be used to perform "what-if" exercises and to identify and assess candidate problem workarounds. Finally, use of the IMP/IMS focuses and strengthens the interaction between the government and contractor teams with respect to program execution.

Best Practices and Lessons Learned

Right type and level of detail. The IMP should provide sufficient definition to track the step-by-step completion of the required accomplishments for each event, and to demonstrate satisfaction of the completion criteria for each accomplishment. Events in the IMP are not tied to calendar dates; they are tied to the accomplishment of a task or work package as evidenced by the satisfaction of the specified criteria for that accomplishment. The IMS should be defined to the level of detail necessary for day-to-day execution of the program.

To build a reasonable IMP and IMS, you need to estimate the attributes of work products and tasks, determine the resources needed, estimate a schedule, and identify and analyze program risks. Accomplishments in the IMP should have criteria for determining completion with clear evidence so that the entire program team can understand the progress. The IMS and IMP should be traceable to the work breakdown structure (WBS) and be linked to the statement of work and ultimately to the earned value management system (EVMS). The WBS specifies the breakout of work tasks that the IMP and

IMS should be built on and the EVMS should report on. A good WBS includes key work efforts partitioned into discrete elements that result in a product (i.e., document, software item, test completion, integrated product) or in measurable progress (percent complete is not recommended when the end-state is not completely quantifiable—an issue in software development, test procedures, or training materials). With a good WBS foundation, both the IMP and IMS can be more useful tools; with the IMP integrating all work efforts into a defined program plan, and the IMS summarizing the detailed schedule for performing those work efforts. The IMP is placed on contract and becomes the baseline execution plan for the program/project. Although fairly detailed, the IMP is a relatively top-level document compared to the IMS. The IMS should not be placed on contract; it is normally a contract deliverable.

For evaluating a proposed IMS, focus on realistic task durations, predecessor/successor relationships, and identification of critical path tasks with viable risk mitigation and contingency plans. An IMS summarized at too high a level often results in obscuring critical execution elements and contributing to the EVMS's failure to accurately report progress (for more on EVMS, see the SEG's "Acquisition Management Metrics" article). A high-level IMS may also fail to show related risk management approaches being used, which often results in long-duration tasks and artificial linkages masking the true critical path.

An example of this is an IMS with several concurrent activities progressing in parallel and showing a critical path along one activity that later links and transitions to another activity. A third activity also shows a dependency to the first, but it is not considered to be on the critical path. If the IMS does not have the detail to determine the progress point at which the critical path transitions to the second activity, the real critical path could be along the dependency of the third activity. Conversely an IMS that is too detailed may result in similar problems; the critical path is too hard to identify (looking in the weeds not up at the trees). The IMS's physical maintenance becomes tedious and linkages could be missed in the details. An IMS can be ineffective on a program when it is either too high level or too detailed.

In general, the IMP can be thought of as the top-down planning tool and the IMS as the bottom-up execution tool for those plans. Note, however, that the IMS is a scheduling tool for management control of program progression, not for cost collection purposes.

Measurable criteria. Criteria established for IMP accomplishments should be measurable (i.e., satisfactory completion of a test event, approval of a study report, or verification of an activity or test). Consider including accomplishment of critical performance requirements (key performance parameters or technical performance metrics). For these, it important to link criteria to the specification versus the actual performance requirement embedded in the criteria so requirements do not have to be maintained in more than one document.

Multiple delivery/increment programs. On programs with multiple deliveries and/or multiple increments, ensure that the IMS includes cross-delivery order and cross-increment relationships. This is valuable when conducting critical path analyses on the IMS. These relationships sometimes drive "ripple effects" across the delivery orders and work tasks, and when analyzing a critical path or estimating a "what if" or total cost for a modification, this is an extremely valuable factor.

Stakeholder involvement. Relevant stakeholders (including user organizations, financial managers, and sustainment organizations) should be involved in the planning process from all life-cycle phases to ensure that all technical and support activities are adequately addressed in program plans such as the IMP and IMS.

Communicating via IMS. The IMS can be used to communicate with stakeholders on a regular basis. For enterprise systems with large numbers of external interfaces and programs, the IMS can be used as the integration tool to indicate and track milestones relevant to the other programs.

References and Resources

Acquisition Community Connection, Integrated Master Plan (IMP)/Integrated Master Schedule (IMS).

AFMC Pamphlet 63-5, November 11, 2004, Integrated Master Plan and Schedule Guide.

Department of Defense, October 21, 2005, *Integrated Master Plan and Integrated Master Schedule: Preparation and Use Guide*, ver. 0.9.

ACQUISITION PROGRAM PLANNING

Performance Engineering

Definition: *Performance engineering is a specialty systems engineering discipline that encompasses the practices, techniques, and activities required during each phase of the Systems Development Life Cycle [1] to ensure that a proposed or existing solution will meet its nonfunctional requirements. Nonfunctional requirements specify the criteria used to judge the operation of a system rather than its specific behaviors or functions.*

Keywords: *capacity planning, design validation, feasibility, instrumentation, load testing, measurement, modeling and simulation, monitoring, requirements validation, response time, scalability, stress testing, throughput*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of performance engineering in the acquisition process, where it occurs in systems development, and the benefits of employing it. MITRE SEs are also expected to understand and recommend when performance engineering is appropriate to a situation. Some aspects of performance engineering are often associated with specialty engineering disciplines. Others, however, are the purview of mainstream systems and design engineering (e.g., many of the dimensions of usability). MITRE SEs are expected to monitor and evaluate performance engineering technical efforts and the acquisition program's overall performance engineering activities and recommend changes when warranted, including the need to apply specialty engineering expertise.

Performance Engineering Scope

Performance engineering focuses on the ability of systems to meet their nonfunctional requirements. A nonfunctional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. It may address a property the end product must possess, the standards by which it must be created, or the environment in which it must exist. Examples are usability, maintainability, extensibility, scalability, reusability, security and transportability. Performance engineering activities occur in each phase of the Systems Development Life Cycle. It includes defining nonfunctional requirements; assessing alternative architectures; developing test plans, procedures, and scripts to support load and stress testing; conducting benchmarking and prototyping activities; incorporating performance into software development; monitoring production systems; performing root cause analysis; and supporting capacity planning activities. The performance engineering discipline is grounded in expertise in modeling and simulation, measurement techniques, and statistical methods.

Traditionally, much of performance engineering has been concerned with the performance of hardware and software systems, focusing on measurable items such as throughput, response time, and utilization, as well as some of the "-ilities"—availability, reliability, scalability, and usability. Tying the performance of hardware and software components to the mission or objectives of the enterprise should be the goal when conducting performance engineering activities. This presents performance results to stakeholders in a more meaningful way.

Although performance engineering activities are most often associated with hardware and software elements of a system, its principles and techniques can be applied to other aspects of systems that can be measured in some meaningful way, including, for example, business processes. In the most simplistic sense, a system accepts an input and produces an output. Therefore, performance engineering is applicable to not only systems but networks of systems, enterprises, and other examples of complex systems.

As an example, given the critical nature of air traffic control systems, their ability to meet nonfunctional requirements, such as response time and availability, is vital to National Airspace System (NAS) operations. Though there are many air traffic control systems within the NAS, the NAS itself is an example of an enterprise comprising people, processes, hardware, and software, among other things. At any given time, the NAS has a finite capacity; however, an opportunity exists to increase that capacity through more efficient processes or new technology. The NAS is an example of a non-IT system to which performance engineering techniques can be applied.

Performance Engineering Across the Systems Engineering Life Cycle

As illustrated in Figure 1, the activities associated with performance engineering span the entire systems life cycle—from Pre-Systems Acquisition through Sustainment. Although performance engineering is recognized as fundamental in manufacturing and production, its activities should begin earlier in the system life cycle when an opportunity exists to influence the concept or design to ensure that performance requirements can be met. Performance engineering techniques can be used to determine the feasibility of a particular solution or to validate the concept or requirements in the Pre-Systems Acquisition stage of the life cycle. Likewise, performance engineering techniques can be used to conduct design validation as well.

Performance Engineering Activities

Performance engineering includes various risk reduction activities that ensure that a system can meet its nonfunctional requirements. Performance engineering techniques can be used to validate various aspects of a planned system (whether new or evolving). For instance, performance engineering is concerned with validating that the nonfunctional requirements for a particular system are feasible even before a design for that system is in place. In this regard,

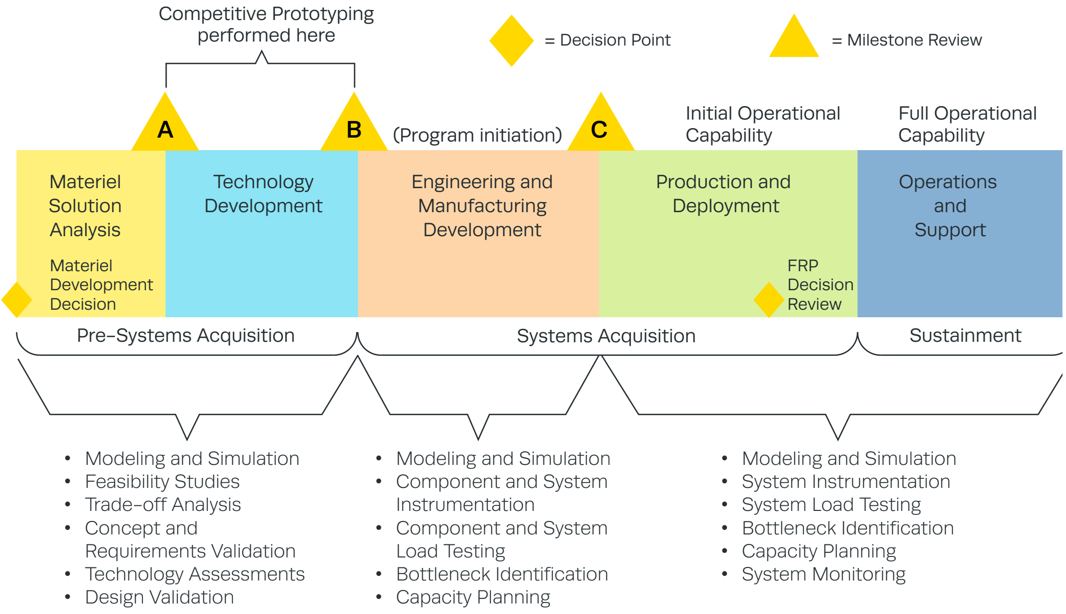

Competitive Prototyping performed here
= Decision Point
= Milestone Review
A
B
(Program initiation)
C
Initial Operational Capability
Full Operational Capability
Materiel Solution Analysis
Materiel Development Decision
Technology Development
Engineering and Manufacturing Development
Production and Deployment
FRP Decision Review
Operations and Support
Pre-Systems Acquisition
Systems Acquisition
Sustainment
• Modeling and Simulation
• Feasibility Studies
• Trade-off Analysis
• Concept and Requirements Validation
• Technology Assessments
• Design Validation
• Modeling and Simulation
• Component and System Instrumentation
• Component and System Load Testing
• Bottleneck Identification
• Capacity Planning
• Modeling and Simulation
• System Instrumentation
• System Load Testing
• Bottleneck Identification
• Capacity Planning
• System Monitoring

Figure 1. Performance Engineering in the System Life Cycle

requirements validation ensures that the nonfunctional requirements, as written, can be met using a reasonable architecture, design, and existing technology.

Once a design is in place, performance engineering techniques can be used to ensure that the particular design will continue to meet the nonfunctional requirements prior to actually building that system. Design validation is a form of feasibility study used to determine whether the design is feasible with respect to meeting the nonfunctional requirements. Likewise, performance engineering activities can be used, as part of a technology assessment, to assess a particular high-risk aspect of a design.

Finally, trade-off analysis is related to all of the activities mentioned previously in that performance engineering stresses the importance of conducting a what-if analysis—an iterative exploration in which various aspects of an architecture or design are traded off to assess the impact. Performance modeling and simulation as well as other quantitative analysis techniques are often used to conduct design validation as well as trade-off, or what-if, analyses.

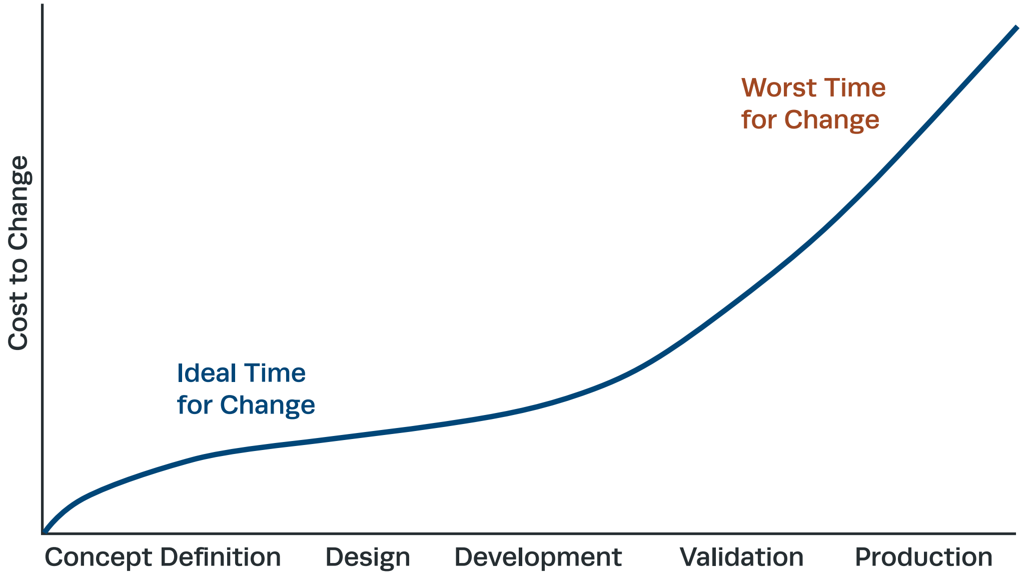
Once a system is deployed, it is important to monitor and measure function and performance to ensure that problems are alleviated or avoided. Monitoring a system means being aware of the system's state in order to respond to potential problems. There are different levels of monitoring. At a minimum, monitoring should reveal whether a particular system component is available for use. Monitoring may also include the collection of various measurements such as the system load and resource utilization over time. Ideally availability and measurement data collected as part of the monitoring process are archived in order to support performance analysis and to track trends, which can be used to make predictions about the future. If a permanent measuring and monitoring capability is to be built into a system, its impacts on the overall performance must be taken into consideration during the design and implementation of that system. This is characterized as measurement overhead and should be factored into the overall performance measurement of the system.

System instrumentation is concerned with the measurement of a system, under controlled conditions, to determine how that system will respond under those conditions. Load testing is a form of system instrumentation in which an artificial load is injected into the system to determine how the system will respond under that load. Understanding how the system responds under a particular load implies that additional measurements, such as response times and resource utilizations, must be collected during the load test activity as well. If the system is unable to handle the load such that the response times or utilization of resources increases to an unacceptable level or shows an unhealthy upward trend, it may be necessary to identify the system bottleneck. A system bottleneck is a component that limits the throughput of the system and often impacts its scalability. A scalable system is one whose throughput increases proportionally to the capacity of the hardware when hardware is added. Note that elements like load balancing components can affect the proportion by which capacity can be increased. Careful planning is necessary to ensure that analysis of the collected data will reveal meaningful information.

Finally, capacity planning is a performance engineering activity that determines whether a system is capable of handling increased load that is predicted in the future. Capacity planning is related to all the activities mentioned previously—the ability to respond to predicted load and still meet nonfunctional requirements is a cornerstone of capacity planning. Furthermore, measurements and instrumentation are necessary elements of capacity planning. Likewise, because bottlenecks and nonscalable systems limit the capacity of a system, the activities associated with identifying bottlenecks and scalability are closely related to capacity planning as well.

Best Practices and Lessons Learned

System vs. mission performance. The ability to tie the performance of hardware or software or network components to the mission or objectives of the enterprise should be the goal. This allows the results of performance engineering studies to be presented to stakeholders in a more meaning–ful way. It also serves to focus testing on out–comes that are meaningful. For example, central



"Concurrent Engineering," J.R. Hartley, Productivity Press

Figure 2. The Cost of Change

processing unit utilization by itself is not meaningful unless it is the cause of a mission failure or a significant delay in processing critical real-time information.

Early life-cycle performance engineering. Too often, systems are designed and built without doing the early performance engineering analysis associated with the Pre-Systems Acquisition stage shown in Figure 1. When performance engineering is bypassed, stakeholders are often disappointed and the system may even be deemed unusable. Although it is common practice to optimize the system after it's built, the cost associated with implementing changes to accommodate poor performance increases with each phase of the system's life cycle, as shown in Figure 2. Performance engineering activities should begin early in the system's life cycle when an opportunity exists to influence the concept or design of the system in a way that ensures performance requirements can be met.

Risk reduction. Performance engineering activities are used to validate that the nonfunctional requirements for a particular system are feasible even before a design for that system is in place, and especially to assess a particular high-risk aspect of a design in the form of a technology assessment. Without proper analysis, it is difficult to identify and address potential performance problems that may be inherent to a system design before that system is built. Waiting until system integration and test phases to identify and resolve system bottlenecks is too late.

Trade-off analysis. Performance engineering stresses the importance of conducting a trade-off, or what-if, analysis—an iterative analysis in which various aspects of an architecture or design are traded off to assess the impact.

Test-driven design. Under agile development methodologies, such as test-driven design, performance requirements should be a part of the guiding test set. This ensures that the nonfunctional requirements are taken into consideration at all phases of the engineering life cycle and not overlooked.

Monitoring, measurement, and instrumentation. System instrumentation is a critical performance engineering activity. Careful planning is necessary to ensure that useful metrics are specified, that the right monitoring tools are put in place to collect those metrics, and that analysis of the collected data will reveal meaningful information.

Performance challenges in integrated systems. Projects that involve off-the-shelf components or systems of systems introduce special challenges for performance engineering. Modeling and simulation may be useful in trying to anticipate the problems that arise in such contexts and to support root cause analysis should issues emerge/materialize. System instrumentation and analysis of the resulting measurements may become more complex, especially if various subsystems operate on incompatible platforms. Isolating performance problems and bottlenecks may become more difficult as a problem initiated in one system or subsystem may emerge as a performance issue in a different component. Resolving performance engineering issues may require cooperation among different organizations, including hardware, software, and network vendors.

Predicting usage trends. Performance data collected as part of the monitoring process should be archived and analyzed on a regular basis in order to track trends, which can be used to make predictions about the future.

References and Resources

1. Systems Development Life Cycle, Wikipedia, accessed February 5, 2014.

Additional References and Resources

Professional Bodies

Association for Computing Machinery (ACM) [SIGSIM and SIGMETRICS]. Contains special interest groups in both simulation and measurement.

Computer Measurement Group (CMG). A professional organization of performance professionals and practitioners. The CMG holds a yearly conference and publishes a quarterly newsletter.

International Council on Systems Engineering (INCOSE). Recognizes performance engineering as part of the system life cycle.

The Society for Modeling and Simulation International (SCS). Expertise in modeling and simulation, which is used extensively in performance engineering activities.

Performance-Related Frameworks

Capability Maturity Model Integration (CMMI). CMMI is a process improvement approach that provides organizations with the essential elements of effective processes.

Federal Enterprise Architecture (FEA) Performance Reference Model. The PRM is a “reference model” or standardized framework to measure the performance of major IT investments and their contribution to program performance.

Information Technology Infrastructure Library (ITIL). Industry standard for IT service management (includes aspects of performance engineering).

Standards Organizations

Object Management Group (MARTE Profile). International, not-for-profit, computer industry consortium focused on the development of enterprise integration standards. MARTE is “Modeling and Analysis of Real-time and Embedded Systems.”

The Standard Performance Evaluation Corporation (SPEC). A standards body for performance benchmarks. SPEC is an umbrella organization encompassing the efforts of the Open Systems Group.

Transaction Processing Performance Council (TPC). The Transaction Processing Performance Council defines transaction processing and database benchmarks and delivers trusted results to the industry.

Authors

Gunther, N. Author of several performance engineering books.

Jain, R. Professor at Washington University in St. Louis. Author of several performance engineering books and articles.

Maddox, M. 2005, A Performance Process Maturity Model, *MeasureIT*, Issue 3.06.

Menasce, D. A. Professor at George Mason University. Author of several performance engineering books.

Smith, C. U., and L. G. Williams. Creators of the well-known Software Performance Engineering (SPE) process and associated tool. Authors of “Performance Solutions” as well as numerous white papers.

Definition: *Investment analyses take many different forms within the federal government and are applied to support a number of key decisions throughout the investment, acquisition, and program life cycles. Examples include Analysis of Alternatives, Business Case Analysis, Cost-Benefit Analysis, and Economic Analysis.*

Keywords: *analysis of alternatives, business case analysis, cost-benefit analysis, economic analysis, financial analysis, investment analysis*

ACQUISITION PROGRAM PLANNING

Comparison of Investment Analyses

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand key aspects of investment analyses to be performed at various stages of the life cycle, including decisions supported, primary objectives, and general approaches that can be applied. SEs frequently team with economic/cost analysts to perform investment analyses, and their familiarity with key aspects of prevalent analytic approaches can improve the overall quality of completed analyses and the efficiency of supporting activities performed.

Background

Several different analysis approaches are designed to inform sponsor funding and expenditure decisions. These include Analysis of Alternatives, Business Case Analysis, Cost Benefit Analysis, and Economic Analysis. This article provides an overview of the commonalities, overlaps, and differences among prevalent business, investment, and economic analysis approaches and of which analyses are required by different government organizations.

Comparison of Key Investment Analysis Types

- **Analysis of Alternatives (AoA):** An AoA is a technical assessment using distinct metrics and various decision criteria to objectively evaluate different potential courses of action (or alternatives). Typically the focus is on an analysis of alternative technical approaches for meeting a given set of functional requirements or mission needs. Alternatives can, in special cases, be distinguished by different acquisition-related approaches if a real range of feasible technical alternatives does not exist. The assessment also includes a life-cycle cost estimate (LCCE) and risk assessment for each alternative and a preliminary recommendation(s).
- **Business Case Analysis (BCA):** A BCA is used to determine if a new approach and overall acquisition should be undertaken. A BCA provides justification for investment (either to invest or not) based on a comparison of life-cycle costs, benefits, and the results of applying financially oriented calculations such as return on investment (ROI), net present value (NPV), and discounted payback period (DPP) for each alternative. An AoA establishes the alternatives to be evaluated. Based on the results of the financial analysis, a BCA helps determine if a potential new acquisition is warranted and if the effort should proceed.
- **Cost Benefit Analysis (CBA):** The primary objective of a CBA is to identify and obtain approval for the optimum way to solve a specific problem or capitalize on a specific improvement opportunity. A CBA documents the predicted effects (financial, operational, quantitative, and qualitative) of actions under consideration and describes the potential financial impacts and other business benefits to support a a decision regarding adopting a particular solution before significant funds are invested. Agency-specific policies may differ, and CBAs within some organization may need to include a detailed budget estimate and identification of funding sources for the preferred alternative. Many organizations consider a CBA a living document that should be updated as needed.
- **Economic Analysis (EA):** An EA is a systematic approach to identifying, analyzing, and comparing costs and benefits of alternative courses of action. An EA should emphasize an identification of key variables that drive cost, benefit, risk, and uncertainty

assessment results across competing alternatives. Depending on agency policy, an EA may be very similar to a CBA. In some government organizations, EAs will include post-investment management considerations, including the identification of key performance measures for monitoring and evaluating whether initiatives ultimately achieve expected or desired results.

AoAs and BCAs are typically conducted early in the acquisition process for a new information technology (IT) initiative. For example, the Department of Defense (DoD) Joint Capabilities Integration and Development System (JCIDS) process identifies that they should be conducted prior to Milestone A. For each subsequent year, they should be validated against the baseline as costs, schedule, etc., may change.

In the past, many civilian agencies were not necessarily required to prepare distinct AoAs or BCAs but would, instead, rely on the Alternatives Analysis that was prepared in support of an Office of Management and Budget (OMB) Exhibit 300 submission, as required by OMB Circular A-11, Part 7. Although the E-300 is a fairly comprehensive document, a typical AoA or BCA will be more detailed and rigorous.

The DoD requires distinct and more comprehensive AoAs and BCAs primarily because DoD initiatives are often larger dollar investments than civilian IT initiatives. Civilian agencies must periodically check progress against the baseline and, as necessary, revise the E-300 annually. Most recently, a number of civilian agencies (e.g., Department of Homeland Security) have produced guidance for preparing AoAs or BCAs to support investment decision making.

AoAs and BCAs are often well understood among IT acquisition professionals. The understanding of CBAs and EAs is often less clear. The terminology of these analyses is often used interchangeably depending on a given agency and individual's opinion. CBA and EA are basically similar analysis approaches.

Table 1 summarizes the general similarities and differences of the analytical approaches across various agencies/communities.

Best Practices and Lessons Learned

Many identified best practices and lessons learned for particular investment analyses will also be relevant for other investment analysis types.

Analysis of Alternatives (AoA)

Develop viable alternatives. Alternative courses of action that could feasibly be pursued to achieve the mission should be developed. An analysis should not use "throwaway" alternatives to justify a preconceived course of action.

Examine three or more alternatives in addition to any Status Quo (SQ) baseline. For example, new alternatives for traditional software development initiatives might include

Table 1. Comparison of Analytical Approaches

Element	Alternatives Analysis	AoA	BCA	CBA	EA
To address a gap, should I invest or not?		x	x		
I'm going to invest to address a gap. So how should I invest?	x	x	x	x	x
Operational effectiveness		x	x	x	x
LCCE	x	x	x	x	x
Qualitative cost assessment		x	x	x	x
Quantitative benefits assessment	x		x	x	x
Qualitative benefits assessment			x	x	x
ROI calculation	x		x	x	x
Uncertainty analysis	x		x	x	x
Risk analysis		x	x	x	x
Sensitivity analysis			x	x	x
Implementation description		x	x		

commercial-off-the-shelf (COTS), custom software development, and COTS/software (SW) development hybrid. To meet certain requirements, it may be impractical to use all COTS or to solely embark on a custom software development. A well-developed AoA should include technically distinct, realistic alternatives. The SQ alternative should be viable; in particular, SQ does not necessarily mean "do nothing." For example, inclusion of technology refresh in the future may be necessary if it is essential to maintaining required functionality.

Alternatives need to be developed in collaboration with technical subject matter experts (SMEs). An AoA/BCA author should not invent alternatives for the purpose of checking a box and meeting a capital planning requirement. Technical SMEs should be consulted to think through the implications of how to best fulfill mission needs.

Alternatives can be differentiated by acquisition approaches. In certain cases, there may be few distinct technical approaches for fulfilling mission needs.

Establish evaluation criteria and scoring. Identify multiple, independent metrics and compare how each alternative impacts them. Metrics may be quantitative or qualitative. Every metric does not need to be equally important. Weightings can be applied to reflect priorities.

Identify technical and cost risks. Risks with each alternative should be identified, including ease of implementation, relative difficulty to meet certain challenging requirements, as well as an assessment of the probability that an alternative can meet cost and schedule goals.

Life-cycle cost estimating. An AoA should include a multiyear life-cycle estimate generally broken out by acquisition, operations and support

(O&S), and SQ phase-out costs. A typical DoD approach is to analyze O&S costs after 10 years post-system full operational capability. This would include at least one tech refresh cycle and enable the analysis to calculate financial metrics and benefits for each alternative over a longer time period.

Recommendations that consider environmental changes. It is useful to recommend a specific alternative to pursue, and this can be done in conjunction with a subsequent BCA, as needed. In certain cases, it is useful to present a few recommendations that take into account likely changes to the surrounding environment. For example, Alternative 1 may be the preferred recommendation if the program receives adequate funding but not if its funding is cut. Recommendations for specific alternatives can be tied to potential future situations that may arise (which should also be documented in the ground rules and assumptions).

Government/sponsor buy-in. An AoA is sometimes viewed as a required check-the-box activity that is a necessary part of capital planning. For programs requiring large expenditures of funds, however, an AoA is a useful means for the program office to fully consider what must be done to move forward in an effective and efficient manner. The approach requires a degree of rigor that will help the program succeed in the long term.

Business Case Analysis (BCA)

Identify potential benefits for each alternative.

In contrast to an AoA, a BCA identifies measurable benefits for each alternative. An example of a measurable, or quantitative, benefit includes an increase in productivity. Although qualitative benefits such as morale, better information sharing, and improved security cannot be readily measured, they may still be important to alternative selection.

Develop ROI statistics. Key financial metrics help gauge the ROI of alternative courses of action relative to the SQ alternative. No single calculation can fully describe ROI, and financial metrics typically calculated to support this description include net present value (NPV), rate of return (ROR), and discounted payback period (DPP). These three metrics take into consideration the "time value of money" (i.e., a dollar received today is worth more than a dollar received one year from now due to factors such as inflation). All calculations, except NPV, are typically performed relative to the SQ alternative. In other words, the calculations take into account how significantly an alternative course of action increases or decreases costs and benefits relative to the SQ. The NPV for an alternative describes current and future net expected benefits (i.e., expected benefits less expected costs) over the life cycle for that alternative. ROR, stated as a percentage, describes the discount rate at which the incremental current and future benefits of an alternative in comparison to the SQ equal the incremental costs of that alternative relative to the SQ baseline. DPP, stated in units of time, describes how much time it will take for the cumulative incremental benefits of an alternative course of action relative to the SQ alternative to exceed the cumulative incremental costs of an alternative course of action.

Clarify the acquisition/implementation timeline. A clear timeline is important to see how costs are phased in over an investment and O&S period. Programs typically have from two- to five-year investment periods depending on the size of the initiative and the type of acquisition approach.

Identify the risks and potential benefit impact. This pertains specifically to the risks that the LCCE will deviate from the baseline estimate. Specific major cost drivers that are risky should be called out. Software development labor is particularly prone to potential risk of under-estimation. There are others. The risks that certain benefits will not be realized should also be identified. This is similar in concept to identifying cost risks.

Cost Benefit Analysis (CBA)

Alternatives may differ from those in the AoA and BCA and be analyzed in more detail. Depending on how much detail was provided in the AoA and BCA, a CBA should break down the preferred alternative into more specificity. A CBA may focus on how best to implement the preferred alternative, which requirements may yield the biggest "bang for the buck" in higher ROI earlier in the life cycle, how different contracting approaches may reduce risk, etc.

Can analyze multiple alternatives. The standard rule of thumb for a BCA is three new alternatives plus the SQ (policies on the number of alternatives to evaluate may vary, and the number may be predicated on the problem being addressed). A CBA can offer many more, if necessary, with minor variations among each to help determine the best specific approach forward. It will still follow the basic format of a BCA, although the analysis for each alternative approach may not be as comprehensive as in the AoA/BCA.

Allows for incremental ROI results. A CBA is particularly useful for demonstrating the quantitative benefits that specific actions within a work breakdown structure (WBS) may yield. Therefore, a CBA is well suited to analyze if one alternative offers a greater ROI than a competing alternative. Incremental financial analysis helps decision makers move forward by considering which alternative approach at a given point along an integrated master schedule (IMS) will yield a greater "bang for the buck" to justify an investment decision at a particular acquisition milestone or increment.

Economic Analysis (EA)

Perform economic sensitivity analyses for key variables. EAs are often evaluated over the investment life cycle, and it is often unreasonable to assume perfect knowledge of what costs, benefits, and risks will be at all points in the future. For analysis variables that significantly drive analysis results and for which there is either (a) considerable uncertainty or (b) considerable impact if unlikely conditions actually occur, sensitivity analyses or uncertainty-weighted assessments should be performed to determine the potential impact of uncertainty on analysis results.

Evaluate all significant economic implications, but not necessarily in monetary terms. In the federal government, many investment costs, benefits, and risks are not readily translated into monetary terms. Regardless, if these implications are critically important to informing investment decisions, they should be evaluated. Analysts should consider whether other quantitative methods (e.g., comparison of numeric ratings for aspects of investment options) can be applied to assess the relative strengths and weaknesses of investment options.

References and Resources

AFMAN 65-506, "Economic Analysis," August 29, 2011, U.S. Air Force.

Army Cost Benefit Guide, April 8, 2011, Office of the Deputy Assistant Secretary of the Army.

Buck, K. "Conducting a Business Case Analysis," The MITRE Corporation.

Buck, K. "Economic Analysis Documentation," The MITRE Corporation.

OMB Circular A-11, Part 7 ("Planning, Budgeting, Acquisition, and Management of Capital Assets"), Exhibit 300.

Source Selection Preparation and Evaluation

Definition: *Source selection is a critical phase of the pre-award procurement process. It has been thoroughly discussed in regulations and procurement literature. Source selection is often thought of as making trade-offs among offerors' proposals to determine the best value offer.*

Keywords: *advisory multi-step process, best value determination, down-select, evaluation, proposal evaluation, source selection, technical evaluation*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to create technical and engineering portions of request for proposal (RFP) documentation (requirements documents, statement of work, evaluation criteria), assist in developing the technical portions of source selection plans, and assist in the technical evaluation of bidders. MITRE SEs also are expected to encourage agency program and acquisition managers to build effective processes into their acquisition strategies. Increasing the program office's likelihood of success often requires acting as an intermediary between the government and contractors to objectively and independently assess the degree to which proposed solutions or courses of action will provide the capabilities needed to meet the government's needs. This includes conducting assessments of the risk inherent in proposed solutions—including strategies for acquiring (or implementing) them—and identifying actionable options for mitigating those risks.

Background

Source selection has been thoroughly discussed in regulations and procurement literature. One definition would not give it justice. Here are two of the widely used definitions:

> According to Federal Acquisition Regulation (FAR) 15.3, source selection is the "selection of a source or sources in competitive negotiated acquisitions...The objective of source selection is to select the proposal that represents the best value [1]."
>
> The Department of Homeland Security (DHS) Guide to Source Selection defines source selection as "the process used in competitive, negotiated contracting to select the proposal expected to result in the best value to the Government. By definition, negotiation is contracting without using sealed bidding procedures. It is accomplished through solicitation and receipt of proposals from offerors; it permits discussions, persuasion, alteration of initial positions, may afford offerors an opportunity to review their offers before award, and results in award to the proposal representing the best value to the Government."

Source selection is not an isolated aspect of the acquisition life cycle; instead, it is a key phase of the life cycle shown in Figure 1. In order for source selection to be successful, the precursor phases of the life cycle (need identification, market research, requirements definition, strong acquisition planning, solicitation development, and proposal solicitation) must be completed effectively.

The source selection approach should be captured in a source selection plan. The plan should include the proposal evaluation criteria. Selecting appropriate evaluation factors is one of the most important steps in the entire source selection process. The source selection plan explains how proposals are to be solicited and evaluated to make selection decisions. It defines the roles of the source selection team members. A realistic schedule also should be included in the plan.

The article "Picking the Right Contractor" describes best practices and lessons learned in the pre-proposal and selection process, including ways to involve Industry to improve the likelihood of a better source selection outcome. The article "RFP Preparation and Source Selection" will walk you through the RFP process, typical MITRE systems engineering roles, and the important points of the selection process. Both articles contain best practices and lessons learned for the preparation and evaluation processes.

Best Practices and Lessons Learned

Advocate the right definition of success. Some organizations define "acquisition success" as the awarding of the contract. Once the contract is awarded (without a protest), victory is declared.

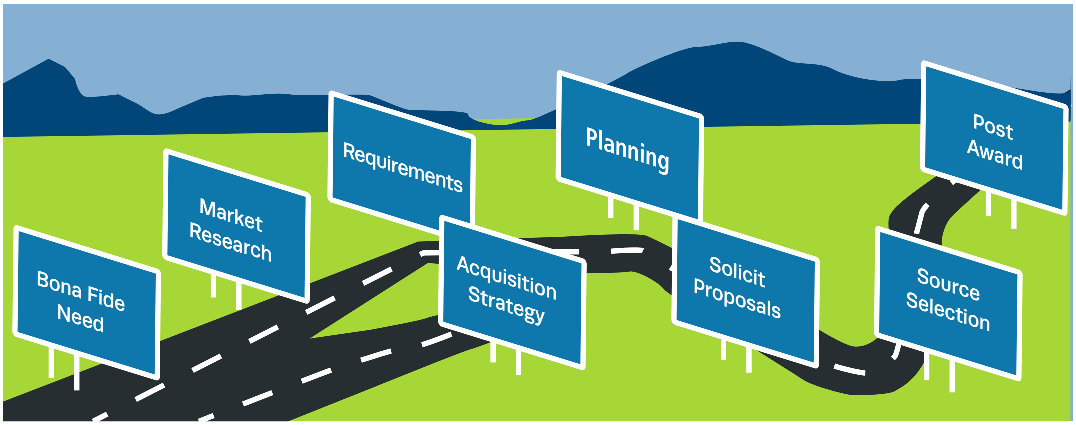


Figure 1. Key Phases of the Acquisition Life Cycle

Although contract award is one of several important milestones, this limited view of acquisition tends to overlook the need to adequately consider what it will take to successfully execute the acquisition effort in a way that achieves the desired outcomes. It leads to a "ready, fire, aim" approach to acquisition planning. Advocate for a broader view of acquisition success, one that balances the desire to award a contract quickly with adequate planning, program management, and systems engineering across the entire system or capability life cycle.

The importance of planning. The importance of conducting adequate acquisition planning before release of the RFP cannot be overstated. This includes encouraging clients to take the time to conduct market research and have dialog with industry so the government becomes a smart buyer that recognizes what is available in the marketplace, including the risks and opportunities associated with being able acquire solutions that meet their needs. This insight allows the government to develop a more effective source selection strategy, which includes choosing more meaningful evaluation factors (or criteria) that focus on key discriminators, linked to outcome metrics. Concentrating on a few key differentiating factors can also translate into a need for less proposal information instead of asking for "everything," which tends to occur when not certain what is important. Adequate acquisition planning helps ensure that the source selection process will go smoothly, increases the probability of selecting the best solution, and reduces the risk of protest.

Maintain the right focus. Focusing on mission/business outcomes instead of detailed technical specifications broadens the trade space of potential innovative solutions that industry (potential contractors) may offer. It can increase industry's ability to use commercial items and/or non-developmental items to fulfill government needs.

Follow your process. The evaluation documentation must provide a strong rationale for the selection decision. During the proposal evaluation phase, a critical lesson is to ensure that the evaluation team does not deviate from the stated RFP evaluation factors. General Accounting Office decisions clearly indicate that use of factors other than those published in the RFP almost guarantees that a bid protest will be sustained. At a minimum, the source selection documentation must identify weaknesses, significant weaknesses, and deficiencies as defined by FAR 15.001 Definitions [1]. Good documentation also identifies strengths and risks.

The importance of industry exchanges. Increased communication with industry through presolicitation notices, information exchanges, and draft RFPs makes the acquisition process more transparent and may lower the likelihood of a protest. These techniques can be an effective way to increase competition, especially when there is a strong incumbent. Exchanges with industry are especially important when the procurement requirements are complex.

Handling sensitive proposal information—a critical requirement. To maintain the integrity of procurement, sensitive source selection information must be handled with discretion to avoid compromise. All government team participants share the critical responsibility to ensure that source selection and proprietary information is not disclosed. There is no room for error. Any lapses by MITRE individuals not only could compromise the integrity of a federal procurement but also could damage MITRE's relationship with the government.

Clarity of evaluation factors. It is not unusual for the government to ask MITRE SEs to help draft proposal evaluation factors (Section M) for a solicitation. The focus should be on the key discriminators that will help distinguish one proposal from another. Cost must always be one of the factors, along with such factors as mission capability, similar experience, past performance, and key personnel. Many solicitations are often vague about the relative weights among such evaluation factors as cost. These ambiguities often lead to successful protests. It is important to do everything possible to ensure that the relative weights of the factors are as clear as possible in the minds of the potential offerors and the government evaluation team.

References and Resources

1. Federal Acquisition Regulation (FAR), http://www.acquisition.gov/far/, accessed February 5, 2014.
   FAR 15.001 Definitions
   FAR 15.1 Source Selection Processes and Techniques
   FAR 15.202 Advisory Multi-step Process
   FAR 15.3 Source Selection

Additional References and Resources

Asset Reuse (Procurement): The Use of Industry Exchange to Increase Competition.

MITRE P&P CR 3.2 Support to Sponsors' Source Selection Proceedings.

MITRE's Input to the Data Access and Dissemination Systems (DADS) Cost Evaluation Lessons Learned, September 25, 2009.

NCMA World Congress, April 22–25, 2007, Source Selection: Best Practices in Streamlining the Process.

SOURCE SELECTION PREPARATION AND EVALUATION

Picking the Right Contractor

Definition: *The Advisory Multi-step Process is a presolicitation process that can help to help define requirements, streamline competition, increase competition, and reduce the likelihood of a protest.*

Keywords: *advisory multi-step process, down-select, draft RFP, exchanges with industry*

MITRE SE Roles and Expectations: In the early planning stages of a major acquisition, MITRE systems engineers (SEs) and acquisition experts are expected to encourage agency program and acquisition managers to build into their acquisition strategies sufficient time to employ the Advisory Multi-step Process in conjunction with Industry Exchanges and draft RFPs. These presolicitation techniques are powerful tools that can help define requirements, streamline competition, increase competition, and reduce the likelihood of a protest. These techniques can be used to increase the probability of a successful source selection.

How the Advisory Multi-step Process Works

The Advisory Multi-step Process is described in the FAR at 15.202 [1]. Generally, in an advisory multi-step process, agencies issue a presolicitation notice inviting potential offerors to submit sufficient information to allow the government to judge whether the offeror could be a viable competitor. The presolicitation notice should identify the information that must be submitted and the criteria that will be used in making the evaluation. Some examples of information include a statement of qualifications, proposed technical concept, past performance, and limited pricing information. The presolicitation notice should contain sufficient information to permit a potential offeror to make an informed decision about whether to participate in the acquisition. Examples include asking for information on specific existing technologies of a maturity to be demonstrable in a laboratory environment, or asking for experience with a new development technique or technology. The information should allow for discrete differentiation between industry respondents that can clearly inform industry of their ability to compete.

The agency must evaluate all responses in accordance with the criteria stated in the notice and must advise each respondent in writing whether it will be invited to participate in the resultant acquisition or, based on the information submitted, that it is unlikely to be a viable competitor. The agency must advise respondents considered not to be viable competitors of the general basis for that opinion. Notwithstanding the results of this initial evaluation, all respondents may participate in the resultant acquisition.

Best Practices and Lessons Learned

A more manageable and efficient source selection process. This multi-step technique has been used to produce a more manageable and efficient source selection process; potential offerors learn early in the process that they may not be able to compete effectively. Industry benefits by avoiding expenditure of unnecessary business development resources. Government benefits by avoiding the expenditure of scarce evaluation resources on weak proposals. This technique can be used to make the evaluation process more manageable and streamlined in a situation where an agency is expecting to receive a large number of offers. On the other hand, the technique has also been used successfully to increase competition when a strong incumbent may be discouraging competition.

In both cases, the likelihood of a protest is reduced when there is clear, frequent, and fair communication with Industry.

More open communication with industry. A logical follow-on to the Advisory Multi-step Process is to conduct a series of information exchanges with those respondents who have been determined to be viable competitors. Before receipt of proposals, the FAR allows for one-on-one interaction with Industry. FAR 15.201(b) states: "The purpose of exchanging information is to improve the understanding of Government requirements and industry capabilities." These one-on-one sessions can be powerful opportunities for open communication. Experience has shown that open forums serve to inform Industry, but do not inform the government of industry capabilities. The normal hesitation to ask the important questions in a large forum evaporates in private sessions when competitors are not in the room. Moreover, the benefits to the government from these sessions include better understanding of requirements and an improved solicitation.

A 2007 NCMA World Congress presentation calls exchanges with industry a best practice for the solicitation phase. One of the critical lessons learned on a major civilian agency acquisition concerned exchanges with industry: "The technical exchange...led to RFP modifications improving the solicitation, and to a better understanding of the Government's requirements [2]."

Changing the competitive environment with a draft RFP. When there is sufficient time in the presolicitation phase of a procurement, agencies should be encouraged to issue draft RFPs as a key attachment to the presolicitation notice issued under the Advisory Multi-step Process. Draft RFPs provide the information needed to help potential respondents make decisions about whether to participate. In addition, draft RFPs inform the industry exchanges described earlier.

Lessons learned from multiple agency procurements indicate that draft RFPs inform industry of the desire for open competition and increase the competitive field, thereby allowing for stronger technical and cost competition and providing clearer direction and requirements for the RFP. Draft RFPs can change the competitive environment of an acquisition, especially when there has been a long-term incumbent.

- Industry exchanges should be used to help the government communicate more clearly, especially when the requirements are complex.
- Multi-step techniques can increase tech nical and cost competition.
- Multi-step techniques are an effective way to increase competition when there is a strong incumbent.
- Pre-solicitation planning has a high payoff for all participants.
- Increasing communication with industry through presolicitation notices, information exchanges, and draft RFPs makes the acquisition process more transparent and lowers the likelihood of a protest.

References and Resources

1. Federal Acquisition Regulation (FAR) 15.202 Advisory Multi-step Process, effective November 13, 2009.
2. Zwiselsberger, D., A. Holahan, and J. Latka, April 25, 2007, "Source Selection: Best Practices in Streamlining the Process," NCMA World Congress, April 22–25, 2007.

SOURCE SELECTION PREPARATION AND EVALUATION

RFP Preparation and Source Selection

Definition: *RFP preparation and source selection are the actions necessary to prepare for a government solicitation and to select one or more contractors for delivery of a product or service.*

Keywords: *acquisitions, competitive procurement, non-competitive procurement, proposal, RFP, RFP development, source selection, strategy*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to create technical and engineering portions of Request for Proposal (RFP) documentation (requirements documents, statement of work, evaluation criteria) and to assist in the technical evaluation of bidders during source selection.

Background

A program's acquisition strategy addresses the objectives for the acquisition, the constraints, availability of resources and technologies, consideration of acquisition methods, types of contracts, terms and conditions of the contracts, management considerations, risk, and the logistics considerations for the resulting products. The acquisition strategy identifies the context for development of the RFP and source selection as either a "competitive" or "non-competitive" procurement. The requirements contained in the RFP, and the contractor(s) selected during a procurement, can often determine the success or failure of a system for the duration of a program. Using a process to develop the RFP and conduct a source selection can significantly improve the likelihood of success. Doing it right the first time is critical—rarely does a program have a chance to do it again.

Competitive Procurement

In a competitive procurement, two or more contractors, acting independently, are solicited to respond to the government RFP. Their proposals are evaluated during source selection,

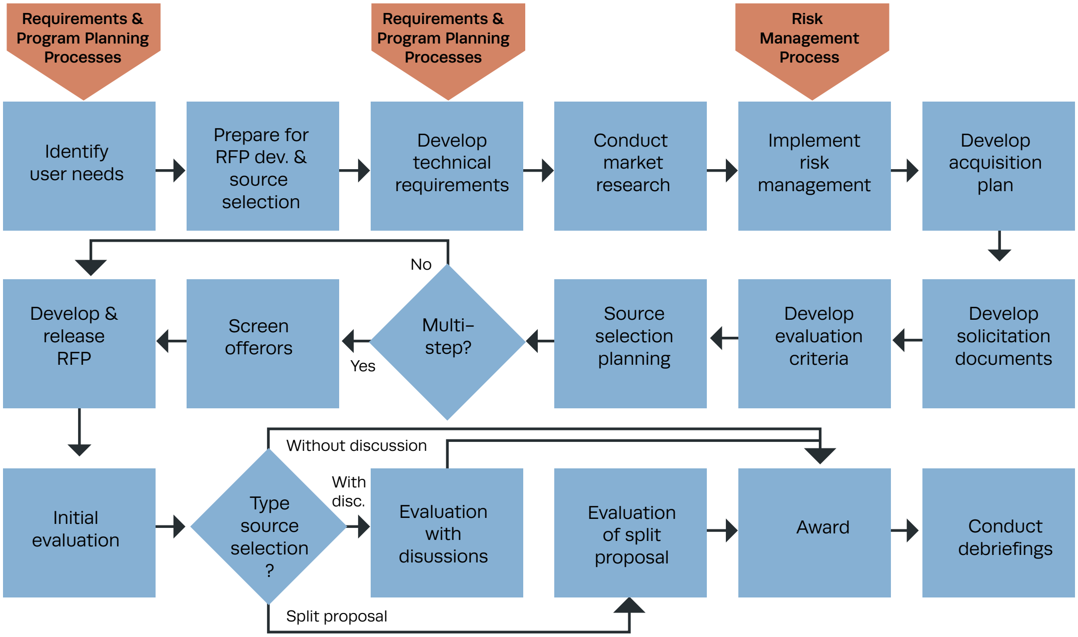

Requirements & Program Planning Processes
Requirements & Program Planning Processes
Risk Management Process
Identify user needs
Prepare for RFP dev. & source selection
Develop technical requirements
Conduct market research
Implement risk management
Develop acquisition plan
Develop & release RFP
Screen offerors
No
Yes
Multi–step?
Source selection planning
Develop evaluation criteria
Develop solicitation documents
Without discussion
With disc.
Initial evaluation
Type source selection ?
Evaluation with dissussions
Evaluation of split proposal
Award
Conduct debriefings
Split proposal

Figure 1. Competitive Procurement Actions

and the contract is awarded to the contractor(s) who offers the most favorable terms to the government.

Competitive procurement activities include the preparation steps that lead to development of the acquisition strategy, which as noted above, provides the basis for developing the RFP and conducting the source selection. Leading up to this are the development of solicitation documents, evaluation criteria, and source selection approach. If the program office wants to reduce the number of contractors proposing, it can conduct a multi-step competitive procurement. One example of this is to use competitive prototyping as a step to further evaluate the qualifications of competing contractors under more representative conditions. Depending on the amount of development required for the program (and under DoD 5000.02), competitive prototyping should not only be recommended but required. The overall activities in a competitive procurement are illustrated in Figure 1.

RFP Development Process

RFP development is part of the overall procurement process. The actions necessary for development and release of the RFP are shown in Figure 2.

The acquisition strategy provides the overall guidance for development of the RFP, and the work breakdown structure (WBS) provides the definition of the program and guides the contractor in creating the contract WBS. The specifications or technical/system requirements document (TRD/SRD), the statement of objectives or work (SOO/SOW), and the contract data

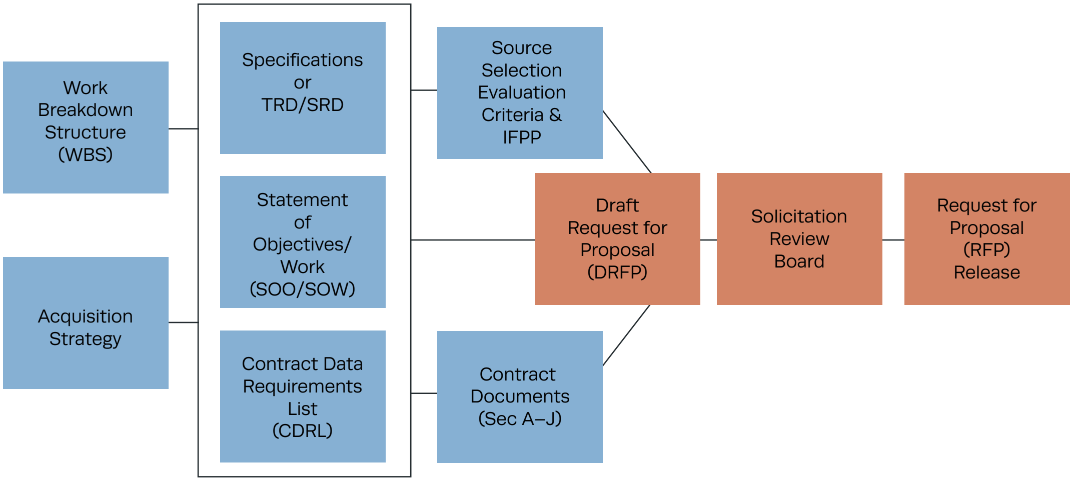

Work Breakdown Structure (WBS)
Acquisition Strategy
Specifications or TRD/SRD
Statement of Objectives/ Work (SOO/SOW)
Contract Data Requirements List (CDRL)
Source Selection Evaluation Criteria & IFPP
Contract Documents (Sec A–J)
Draft Request for Proposal (DRFP)
Solicitation Review Board
Request for Proposal (RFP) Release

Figure 2. RFP Development Actions

requirements list (CDRL) form the technical basis of the RFP. These are usually the focus of MITRE SEs on RFPs.

An RFP is divided into sections A–M. Sections A–J are primarily contract documents, except for section C, which is the SOO or SOW. Section K contains attachments like the TRD/SRD, section L is the Instructions For Proposal Preparation, and section M is the evaluation criteria. MITRE is often asked to participate in the construction of sections L and M. Evaluation criteria are another critical component of the RFP. Technical criteria need to be included and need to address areas of technical risk and complexity.

Source Selection

In a competitive procurement of a system/project, source selection is the process wherein proposals are examined against the requirements, facts, recommendations, and government policy relevant to an award decision, and, in general, the best value proposal is selected. The actions shown in Figure 3 are those generally conducted during source selection. The focus of MITRE's participation in source selections is the evaluation of the technical proposal and the resulting risk assessment.

Non–Competitive Procurement

Although it is not a common initial procurement approach, on occasion non-competitive procurement is necessary to meet government needs for certain critical procurements. This approach is more commonly used with a contractor who is already on contract with the government (but not necessarily the same organization doing the procurement) providing a similar capability, or when it is clearly advantageous to use the non-competitive approach in subsequent contract changes or new solicitations for an existing program.

As with competitive procurement, the actions taken in a non-competitive procurement include the preparation steps that lead to development of the acquisition strategy. Prior to development of the solicitation documents that constitute the RFP, the program office must submit Justification & Approval (J&A) documentation to the appropriate agency office to receive approval for the non-competitive procurement. Occasionally there is a technical reason for using a particular contractor, and MITRE is involved with generating the J&A. With this approval, the program office can develop the solicitation documents and enter into collaborative contract development with the contractor. On completion of the collaborative contract development, the program office evaluates, negotiates, and awards a contract with many of the steps indicated above. MITRE is most often used to evaluate the proposal for technical approach and resources/engineering hours.

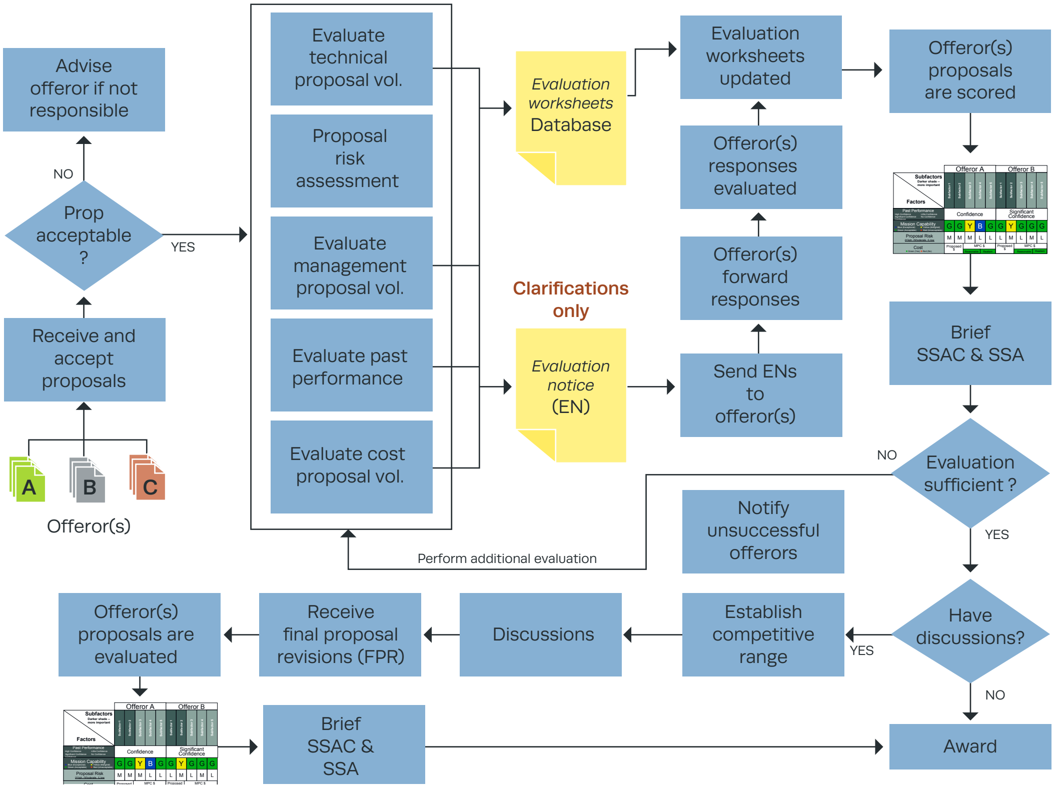

Advise offeror if not responsible
NO
Prop acceptable ?
YES
Receive and accept proposals
A
B
C
Offeror(s)
Evaluate technical proposal vol.
Proposal risk assessment
Evaluate management proposal vol.
Evaluate past performance
Evaluate cost proposal vol.
Evaluation worksheets Database
Clarifications only
Evaluation notice (EN)
Evaluation worksheets updated
Offeror(s) responses evaluated
Offeror(s) forward responses
Send ENs to offeror(s)
Offeror(s) proposals are scored
Brief SSAC & SSA
NO
Evaluation sufficient ?
YES
Notify unsuccessful offerors
Perform additional evaluation
Have discussions?
YES
NO
Establish competitive range
Discussions
Receive final proposal revisions (FPR)
Offeror(s) proposals are evaluated
Brief SSAC & SSA
Award

Figure 3. Source Selection Actions

Best Practices and Lessons Learned

Getting the most bang for your bucks—market research and competitive prototyping. Although it is time-consuming, spending time researching the state of the art and visiting with contractors and vendors will give you a good sense of what's achievable for program requirements. In competitive procurements, solicitations are very helpful in determining the range of available developers/suppliers. Solicitations may also be used to perform work toward the acquisition; meaning asking industry to submit papers and demonstrations prior to the release of an RFP. MITRE, as an operator of FFRDCs, may review this kind of proprietary information and use it as a basis for validating technology or assumptions about requirements. Such feedback from industry may also be useful for refining the evaluation criteria for the RFP.

Competitive prototyping can be used to require competing developers to demonstrate applicable technology or services, along with engineering process and documentation (as examples) to enable better evaluation of their overall abilities to deliver the full program. It may also be used as a technique to reduce risk in complex or unproven technical areas. For more information on competitive prototyping, see the article "Competitive Prototyping" under the SEG's Contractor Evaluation topic.

The right level of detail for a WBS. The WBS is often the foundation used for determining contractor progress and earned value during the program development and deployment phases. As such, it needs to be structured to provide enough detail to judge sufficient progress. WBS elements should be broken down into efforts no larger than 60 days per unit. At least 90+ percent of the WBS elements must be measureable in durations of 60 days or less. This allows you to track WBS completion on a quarterly basis and get a good idea of progress on a monthly basis. Each WBS item should only have three reporting states: zero percent complete (not started), 50 percent complete (at least 50 percent complete), 100 percent complete (done). This allows you to track the WBS status without overestimating the percent complete. It is possible if the development effort is quite large that this level of detail may result in a very large integrated schedule and plan later (see the article "Integrated Master Schedule (IMS)/ Integrated Master Plan (IMP) Application"), but you want the WBS foundation to allow for as much or as little detail as may be applied later on. Keeping the WBS at a high level will definitely cause an inability to judge accurate progress during program execution, and lead to late identification of cost and schedule risks.

What matters in the RFP. Depending on the acquisition strategy chosen, the completeness of the TRD/SRD is critical. Programs expecting to evolve over time through "Agile" or "Evolutionary" acquisition strategies will need to have carefully chosen threshold requirements specified for the initial delivery of capability; ones that are achievable within the allotted schedule for that first delivery. Requirements to be satisfied in a later delivery of capability may be less stringent if they are apt to change before being contracted for development. In a more traditional acquisition strategy where all requirements are to be satisfied in one or two deliveries of capability, the TRD/SRD must be complete.

Another point to remember is that TRD/SRDs and SOO/SOW form the basis of testing—both sets of documents need to be written with a focus on performance and test. Waiting until test preparation is too late to discover that requirements were not stated in a manner that is quantifiable or testable. For more information on requirements and testing, see the System Design and Development and Test and Evaluation topics in the SEG's SE Life-Cycle Building Blocks section.

Evaluation criteria need to be comprehensive and specific enough to allow clear differentiation between offerors, especially on those areas of requirements of critical importance to the success of the program—try a sample proposal against the criteria to see if they are in fact selective enough. There have been cases where the criteria have not been expansive enough and differentiating technical information found in the proposals to be relevant to selection could not be considered for evaluation. Beyond just the written criteria, consider requiring the offerors to provide and follow their risk management process or software development plan as part of an exercise or demonstration.

Source selection—be prepared. MITRE engineers responsible for evaluating technical proposals need to be well versed in applicable current technology for the program. If a proposal contains a new technical approach that is unfamiliar, perform the research to determine the viability. Do not assume the approach is low risk or commonplace; do the research to determine feasibility, risk ,and the proposing contractor's familiarity with it. Consult with MITRE experts in our "tech centers" to provide expertise in areas where program staff are limited in depth of knowledge; getting the right assistance in source selection is critical to choosing the right contractor. You don't get another chance!

The danger of "leveling." During the source selection, offerors are often asked clarification questions in writing, or asked to provide oral proposals and questions/answer sessions. The result of several iterations of these information exchanges among the offerors can result in or look like "leveling;" when the government team has effectively obtained similar information and technical approaches across the offerors, resulting in similar risk and allowing a final selection based purely on cost. Beware that this is usually not the factual result but a perception and result of iterative clarification calls: that all the offerors have provided adequate detail and reasonable technical approaches. As most MITRE SEs who have participated in multiple source selections will tell you, the offerors are not likely to be even in terms of technical risk, past experience/expertise, or in architectural approach. It is up to the engineering team to clarify the differences so that "leveling" does not occur. This means careful consideration of the evaluation criteria for differentiation, focusing on the critical areas needed for success on the program. If the offeror has not demonstrated a consistent approach throughout the proposal process, this in itself may be a legitimate weakness.

Leverage in a sole-source environment. It is a best practice to judge a proposed effort on a sole-source contract against similar past efforts already expensed and for which hours are actual. Again, the ability to do this is dependent on the initial contract (specifically the WBS) being structured to capture progress at an appropriate level to accrue cost and schedule for independent efforts. Lacking a reasonable facsimile for the proposed effort will require either experience in the contractor's development methodology enough to estimate hours or research into other programs and developments to compare against (both inherently less helpful for negotiating).

References and Resources

Air Force Materiel Command (AFMC), November 2004, AFMC Integrated Master Plan and Schedule (IMP/IMS) Guide.

Air Force Materiel Command (AFMC), April 2005, HQ AFMC Justification and Approval Preparation Guide and Template.

Bloom, M., and J. Duquette, July 2006, Systems Engineering in RFP Prep and Source Selection Process V3.0, The MITRE Corporation.

Defense Federal Acquisition Regulation Supplement and Procedures, Guidance, and Information, Contracting by Negotiation, DFAR Subpart 215.

Department of Defense, April 3, 1996, MIL-HDBK-245D DoD Handbook for Preparation of Statement of Work (SOW).

Department of Defense, August 1, 2003, MIL-STD-961E DoD Standard Practice Defense and Program-Unique Specifications Format and Content.

Department of Defense, July 30, 2005, MIL-HDBK-881A DoD Handbook Work Breakdown Structure, Revision A.

Federal Acquisition Regulation, Contracting by Negotiation, FAR Part 15.

IEEE, 1998, IEEE STD 1233 IEEE Guide for Developing System Requirements Specifications.

MITRE SEPO, RFP Preparation and Source Selection Toolkit.

OC-ALC/AE (ACE), June 20, 2003, Statement of Objectives (SOO) Information Guide.

U.S. Air Force, January 2004, Other Than Full and Open Competition, *AFFARS (Air Force Federal Acquisition Regulation Supplement) MP5306.3.*

Program Acquisition Strategy Formulation

Definition: *Developing a comprehensive, integrated acquisition strategy is an acquisition planning activity. It describes business, technical, and support strategies to manage risks and meet objectives. It guides acquisition program execution across the life cycle. It defines acquisition phase and work effort relationships and key program events (decision points, reviews, contract awards, test activities, etc.). It evolves over time and should continuously reflect the program's current status and desired end point. [1].*

Keywords: *acquisition, acquisition strategy, agile acquisition, "big bang" acquisition, contracting, evolutionary acquisition, information technology, software engineering, spirals, systems engineering*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to help our clients/customers craft realistic, robust, and executable acquisition strategies. This means helping articulate what the government needs, translating those needs into mission/outcome-oriented procurement/solicitation requirements, and adequately identifying the issues, risks, and opportunities that shape and influence the soundness of the acquisition strategy. MITRE SEs help agencies achieve what the Federal Acquisition Regulation (FAR) characterizes as "mission-oriented solicitations [2]."

Acquisition Strategy: Answering the Question "How to Acquire?"

Developing and executing an effective acquisition strategy requires "systems thinking" to ensure that the various elements of the strategy are integrated and that interdependencies are understood and accounted for during execution of the strategy. The acquisition strategy is dynamic in that it must reflect changes that occur often during execution. Cost, schedule, and system performance (or capability) trade-offs may also be required, and program managers will need the insight to make informed decisions based on understanding the risks involved in achieving desired outcomes. Therefore, systems engineering plays a key role in both planning and executing this strategy.

How agencies acquire the products and services they need to perform their operations (or execute their mission) varies depending on the criticality or urgency of the product or service to the agency's mission. Another factor is whether the acquisition entails investments or allocation of relatively large amounts of agency resources. Acquisition planning for major, complex efforts requires greater detail and formality, and a greater, earlier need for systems engineering. Acquisition management at this level often requires balancing the equities of multiple stakeholders by establishing and executing a governance process that strikes a balance between the desire for consensus and the agility needed to make trade-offs among cost, schedule, and the capabilities delivered to address changing needs and priorities. Because of the challenges inherent in such an acquisition environment—political, organizational/operational, economic, and technical—the FAR requires program managers to develop and document an acquisition strategy to articulate the business and technical management concepts for achieving program objectives within imposed resource constraints. MITRE SEs are often called on to help develop and execute this strategy.

Best Practices and Lessons Learned

Focus on total strategy. Avoid the temptation to only focus on the contracting aspects of an acquisition strategy. In many agencies, the term "acquisition strategy" typically refers to the contracting aspects of an acquisition effort. This view tends to ignore other factors that influence a successful outcome, including technical, cost or schedule, which often have interdependencies that must be considered to determine how to acquire needed capabilities. The overall strategy for doing so will be shaped and influenced by a variety of factors that must be considered individually and collectively in planning and executing the acquisition effort.

Write it down. The FAR requires program managers to develop an acquisition strategy tailored to the particulars of their program. It further defines acquisition strategy as "the program manager's overall plan for satisfying the mission need in the most effective, economical, and timely manner [3]." An acquisition strategy is not required (or warranted) for all acquisition efforts. However, most (if not all) acquisition efforts for which MITRE provides systems engineering support require acquisition planning and a written plan (acquisition plan) documenting the ground rules, assumptions, and other factors that will guide the acquisition effort. For major system acquisition programs, a written (i.e., documented) acquisition strategy may be used to satisfy FAR requirements for an acquisition plan [4].

Apply early systems engineering. In essence, the FAR requires federal agencies to have increasingly greater detail and formality in the planning process for acquisitions that are more complex and costly. These major or large-scale federal acquisition efforts also have a greater need for application of sound systems engineering principles and practices, both within the government and among the suppliers (industrial base) of the products and services. A well-known axiom of program/project management is that most programs fail at the beginning [5]. In part this can be attributed to inadequate systems engineering, which when done effectively articulates what the government needs, translates those needs into procurement/solicitation requirements, and identifies issues, risks, and opportunities that shape and influence the soundness of the acquisition strategy.

An acquisition strategy is not a single entity. It typically includes several component parts (or strategy elements) that collectively combine to form the overall strategy (or approach) for acquiring via contract(s) the products/supplies or services needed to fulfill an agency's needs. These elements tend to differ depending on the nature of the acquisition effort, whether or not the effort is formally managed as a "program," and the acquisition policies, procedures, and governance policies and regulations of the agency being supported. However, a common element of most acquisition (program) strategies includes structuring the program in terms of how and when needed capabilities will be developed, tested, and delivered to the end user. In general, there are two basic approaches: delivering the capability all at once, in a single step (sometimes referred to as "grand design" or "big bang") to fulfill a well-defined, unchanging need; or delivering capability incrementally in a series of steps (or spirals) to accommodate changes and updates to needs based on feedback from incremental delivery of capability. This latter approach is often referred to as evolutionary or agile acquisition (actual definitions may vary). For more details on each of these strategies, see the articles "Agile Acquisition Strategy," "Evolutionary Acquisition," and "'Big-Bang' Acquisition."

Picking the appropriate strategy. Selecting which capability delivery approach to use depends on several factors, including what is being acquired, the enabling technology's level of maturity, the rate at which technology changes, and the stability of the requirements or evolving nature of the need that the acquisition effort is addressing. For most information technology acquisition efforts, experience has shown that an evolutionary/incremental/agile approach is preferred to accommodate and plan for change inherent in software-intensive systems. However, incorporating an evolutionary or agile approach to delivery of capability is no guarantee that an acquisition strategy is sound. Other factors also determine the effectiveness of a given strategy [5]. These include aspects such as whether the strategy is based on an authoritative assessment of the risk involved in achieving the objectives of the acquisition effort; a realistic test and verification strategy and meaningful event or knowledge–driven milestone reviews to assess progress toward achieving acquisition objectives; and realistic cost, schedule, and performance (delivered capability) baselines.

References and Resources

1. Defense Acquisition Guide Book, accessed February 5, 2014.
2. Federal Acquisition Regulation, Subpart 34.005-2.
3. Federal Acquisition Regulation, Part 34.
4. Federal Acquisition Regulation, Subpart 34.004.
5. S. Meier, Best project management and systems engineering practices for large-scale federal acquisition programs, Acquisition Community Connection Practice Center, DAU.

Definition: *Agile acquisition is a strategy for providing multiple, rapid deliveries of incremental capabilities for operational use and evaluation. Deliveries (called spirals, spins, or sprints) can be a few weeks or months to develop and are built with continuous user participation and feedback. This strategy assumes a predominately software-based development. Do not confuse agile acquisition with agile development, which is "a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams."[1]*

Keywords: *acquisition strategy, agile, agile acquisition, agile development, increment, incremental, uncertainty*

PROGRAM ACQUISITION STRATEGY FORMULATION

Agile Acquisition Strategy

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the conditions under which an agile acquisition strategy will minimize program risk and provide useful capability to customers compared to other strategies. They are expected to recommend agile acquisition for the appropriate situation, and develop and recommend technical requirements, strategies, and processes that facilitate its implementation. MITRE systems engineers are expected to monitor and evaluate program agile acquisition efforts and recommend changes when warranted.

Background

MITRE SEs often are involved in the planning stages of a new program or major modifications to existing programs. As members of the planning team, they must be well acquainted with various acquisition strategies and factors that should be considered when choosing a strategy. Two fundamental reasons for moving toward "agile" processes are:

- To respond quickly to an immediate and pressing need.
- To engage with users throughout the development process and ensure that what is developed meets their needs.

Regardless of whether the final system requirements are clear, agile acquisition strategy is focused on getting capabilities to the user quickly rather than waiting for the final system [2].

Agile acquisition is generally appropriate for two types of system development: 1) enterprise systems with a high degree of functional requirements uncertainty, even if the purpose and intent of the system is known; and 2) small, tactical systems that may have a short life, but an immediate and pressing need. These are cases where the fielding of blocks of an entire system is not feasible because total system requirements cannot be defined at the beginning, but immediate user needs can be partially satisfied with rapidly deployed incremental capabilities. The keys to success of this strategy are close alignment with stakeholders on expectations, and agreement with a limited set of users for providing continuous, rapid user feedback in response to each capability increment.

Advantages of this strategy demonstrate that: 1) development can begin immediately, without the time and expense needed for development, refinement, and approval of functional requirements; and 2) significant user involvement during development guarantees that the capabilities delivered will meet the user's needs and will be used effectively.

Disadvantages are that agile methodologies: 1) usually require stable infrastructure; 2) require significant management and technical oversight to ensure compatibility of ensuing releases; and 3) usually sacrifice documentation and logistics concerns in favor of rapid releases for fielding.

Best Practices and Lessons Learned

When to use agile. The use of agile methods should not be considered for all development efforts. MITRE SEs need to determine if agile methodology is appropriate for the program. It is critical to create evaluation criteria and processes by which any proposed effort can be vetted, to determine if the agile approach is warranted. The minimum criteria are:

- Need for quick initial capability
- High degree of uncertainty in final set of functional requirements

- Fluctuation and complexity due to frequent changes in enterprise business model, data, interfaces, or technology
- Initial architecture or infrastructure that allows for small incremental developments (60–90 day cycles)
- Close cooperation and collaboration with users; identified set of users for continuous interaction with development team

When a contractor bids agile development methodology. If an agile development methodology is proposed by a bidding contractor, determine if the proposed effort is appropriate and has adequate resources in both the contractor and Program Office organizations. For smaller programs, eXtreme Programming techniques may work; for larger efforts requiring scalability, another methodology such as Scrum may work better. In addition, the developing contractor's (and subcontractor's) organization needs to be set up and resourced with the relevant development tools, collaborative processes, and experienced personnel for whichever methodology is to be applied. Government oversight and management of a program using agile methods require staff who are experienced in the methodology and its highly interactive, collaborative management style. In-plant participation by MITRE engineers on a continuous basis should be strongly considered.

Architecture implications. There are differing opinions on the level of effort required to define the overall architecture of an agile activity [2]. Some of this difference is due to the particular development methodology applied. In general, the larger and more complex the system, the greater the effort placed on architecture up front.

Rather than detailing an architecture before development, consider using the first few spirals/sprints to evolve the architecture as more about the system becomes known. This allows the architecture to evolve, based on the needs of the developing system.

The architecture of the system should be designed with enough flexibility so that capabilities can be designed, added, or modified with functional independence. That is, the system is architected and designed so that capabilities scheduled for delivery can interoperate with the components that have been delivered, and do not have critical operational dependencies on capabilities that have not yet been delivered. Layered architectures lend themselves to this application. Components may be developed for different layers; concentrating on the most commonly used/reused components first. An initial capability for delivery could be a component that crosses layers and provides utility to the user ("user-facing").

Lessons learned:

- For larger projects, recognize the need to define business (organization, people, and process) and technical architectures.
- Be aware of agile developers who do not have the skills for designing the architecture, and bring in systems engineers and architects to help.
- Architecturally partition the system into layers and components for clean interfaces and cleanly divided work among

teams. Agile iterations need separable design pieces.

- Identify nonfunctional requirements to build a scalable infrastructure. Large projects will have multiple iterations being worked in parallel. Parallel implementations imply:
    - Discrete implementation units
    - Dependencies between implementation units must be recognized and considered.
- Features delivered for fielding must aggregate into a workable baseline. Dependencies between Configuration Items must be recognized and planned to achieve stable upgrades [3, 4, 5].
- The first few months of an agile development are the most critical time for involvement by MITRE SEs:
    - Lots of collaboration—requires senior people with right skills to provide guidance to developers on priorities and requirements ("stories" or "threads") and to involve the users
    - Skills and levels of expertise may change as different applications and components are being developed. Revisit, as needed.

Infrastructure implications. When agile methodologies are used for developing software systems, a stable infrastructure must be available to the development team and users. A traditionally developed infrastructure provides a stable interface for developers, and allows the agile development team freedom to focus on their section of functionality and respond quickly to changing requirements. If the infrastructure changes often, developers waste time reworking their code to adapt to the new architecture without adding new functionality. Examples include stable operating system interfaces for agile desktop application development and a stable Java Web container architecture for "plug and play" filter implementations [2].

Teaming is key. Constant collaboration between the users and the developers, and among the Program Office, developing organization, and stakeholders, is absolutely critical. Agile methods do not work without this. It is essential that the Program Manager secure user agreement to provide evaluation feedback. It is also necessary that stakeholders be educated on the strategy and benefits of agile methods because agile strategies can be new to government procurement organizations.

Constant communication across the team is essential. Most agile methodologies require daily meetings for status, direction, feedback, and assignments. Meetings are at different levels of the organization—from program management to user feedback sessions. Meetings need to be forums for open communication; problems and limitations need to be identified and addressed openly. Transparency is necessary in planning and development decisions. Keep user evaluation and feedback sessions small, thereby allowing for focused and open communication.

"Quality is job one." Because agile methodologies are about flexibility and change, there can be a fear that quality will suffer in the race to deliver. However, there is an inherent quality driver in agile development if conducted properly. The operational user, who should be constantly evaluating and providing feedback on the product, is an important bellwether of quality. Adherence to development standards, defining segregable capabilities that enable successfully delivered functional increments, constant iterative testing, and constant user feedback all bring quality forward as a critical measurement for agile success.

"Agile" does not mean less disciplined. The various agile development methodologies are quite disciplined. The development teams need to work in parallel within the same development environment without interfering with each other. This requires a disciplined process enforced with constant open communication, good configuration management practices, and quality code to ensure interoperability of the deliveries on the developing baseline. Critical to success is experience of the development team with the agile process and methods. Ensure there are processes in place to enable the communications and tools to support collaboration and parallel development and testing. Quality measurements should be captured and monitored, to include testing successes/failures, defect rates, user comments, and feedback (negative/positive).

References and Resources

1. Wikipedia contributors, "Agile Software Development," January 13, 2010.
2. Dobbins, J. H., S. Luke, A. Epps, R. Case, and J. Wheeler, September 1, 2007, "Agile Acquisition Strategies for NSA Programs," The MITRE Corporation.
3. Doughty, J., September 4, 2007, "Introduction to Agile Development," The MITRE Corporation.
4. Hagan, P., September 4, 2007, "Agile Methods Overview and Implications for Architecture," The MITRE Corporation.
5. Morgan, N., September 4, 2007, "Sponsor Perspective," The MITRE Corporation.

Additional References and Resources

"Agile Acquisition," MITRE Project Leadership Handbook, accessed January 15, 2010.

Coldewey, J., January 19, 2009, The 31 Square-Foot Architecture, Cutter Consortium.

Definition: *Evolutionary acquisition is an acquisition strategy structured to deliver capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on phased definition of capability needs and system requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability over time [1].*

Keywords: *acquisition strategy, capability increments, evolutionary acquisition, operationally useful*

PROGRAM ACQUISITION STRATEGY FORMULATION

Evolutionary Acquisition

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the principles of evolutionary acquisition as it applies to the programs they support. They are expected to evaluate the intended purpose of a proposed acquisition, advise the program manager on the use of an evolutionary acquisition strategy compared to other strategies, and implement that strategy for program planning and execution.

Background

An acquisition strategy is a high-level business and technical management approach that is designed to achieve program objectives within specified resource constraints. It is the framework for planning, organizing, staffing, controlling, and leading a program. It provides a master schedule for research, development, test, production, fielding, and other activities essential for program success. It also provides a master schedule for formulating functional strategies and plans. A primary goal in developing an acquisition strategy is to minimize the time and cost of satisfying an identified, validated need. This goal is consistent with common sense, sound business practices, and the basic policies established by sponsors.

An evolutionary approach delivers capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on consistent and continuous definition of requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability toward a materiel concept [1].

Evolutionary acquisition is a method intended to reduce cycle time and speed the delivery of advanced capability to users. The approach is designed to develop and field mature technologies for hardware and software in manageable pieces. It is focused on providing the operational user with an initial capability that may be less than the full requirement as a trade-off for speed, agility, and affordability. Evolutionary acquisition also allows insertion of new technologies (when they become sufficiently matured) and capabilities over time. In principle, the approach provides the best means of quickly providing advanced technologies to the users while providing the flexibility to improve that capability over time.

There are two approaches to evolutionary acquisition. In the first, the ultimate functionality is defined at the beginning of the program, with the content of each deployable increment determined by the maturation of key technologies. In the second, the ultimate functionality cannot be defined at the beginning of the program, and each increment of capability is defined by the maturation of the technologies matched with the evolving user needs.

Best Practices and Lessons Learned

Architecture is key. This is the most important lesson of this article. To effectively implement evolutionary acquisition, the architecture of the system or capability must be developed first. Employ use-case and similar methodologies to define an "operational architecture" or business process model. Once the operations of the system are understood, a notional system architecture can be generated. The architecture will need to support an evolutionary methodology and should form the basis of the program plan. The first delivered increment should include the target system architecture from which subsequent increments can build. Modeling and simulation may be used to validate architecture assumptions and information flow.

Manage stakeholder expectations and yours, too. Managing expectations ranks high in importance. Whether you define the ultimate functionality up front (not recommended for commercial off-the-self [COTS]-based information technology [IT] systems) or by increment, realistic expectations within the program office and community stakeholders are a must for success. Expect change and expect cost growth. Do not believe you can define the content of all the increments accurately in advance. Change will happen and you need to be prepared. Do not confuse this with "requirements creep;" recognize it as a normal part of the evolutionary process. For more information on managing stakeholder expectations, see the article "Stakeholder Assessment and Management" under the SEG's Transformation Planning and Organizational Change topic.

Understand the operational users' context: Along with managing stakeholder expectations, close coordination and communication with stakeholders is important. User requirements should drive the content and priority of increments, as long as the users understand the end state of the program. This should to be balanced with effective contract execution. Understand the intended mission and operations of the system being acquired; systems engineering in an operational vacuum is not effective. Get acquainted with the operations and the users. Recognize that they have different priorities and a different perspective from a program office. Program offices plan and execute a program; users conduct a mission. It is acceptable to have differing opinions at times. Make sure those differences are handled by open communication leading to positive resolutions.

No technology before its time. Manage technology, instead of technology managing you. The principle behind evolution is to take advantage of emerging technology when it is mature. Stay abreast of new products that can contribute to mission effectiveness, and incorporate routine methods for tracking new technology (e.g., via the annual cycles in the Department of Defense [DoD] for government research laboratories Advanced Technology Demonstrations and Small Business Innovative Research programs). Be aware that new technology sometimes can bring new operational capabilities not previously considered. For example, users see a new toy, and instead of asking for the functionality, they ask for the toy. They see a capability in this toy that you are not aware of and they cannot articulate. Establishing a good working relationship with the stakeholders and users can help you bring out the capability inherent in the toy. This can help mitigate the churn often seen in COTS-based IT systems. Use evolutionary acquisition to your advantage (see the articles "Assessing Technical Maturity" and "Technology Planning"). As long as a useful capability is delivered within a short period (approximately 12–18 months), the users will respond in kind.

Think "parallel developments." Often in an evolutionary model, development of increments must occur in parallel to deliver capability on time. Increments may vary in time to develop and integrate. If done serially, they can extend the program schedule and adversely impact the ability to deliver capability to the users in a timely manner, which was the purpose of evolutionary acquisition. Managing parallel development is challenging but not unachievable; it should not be avoided. Make use of configuration management to control the development baselines and track changes. Allow time in the increment development schedules for the reintegration of a "gold" baseline for final incorporation of parallel changes prior to test and fielding. For more information, see the SEG's Configuration Management topic.

The right contract type. Carefully consider the contract type for an evolutionary acquisition. As evidenced by these lessons learned, changes are frequent and should be part of the plan. Time-and-materials, cost-reimbursement, product-driven payments, or awards can allow for flexibility without severe cost implications. Focus should be on delivery of a useful product, not processes. Indefinite Delivery/Indefinite Quantity style contracts should be considered, but need to be structured so that each delivery order is not treated as an independent entity from the total program (this was seen on a DoD program and was quite painful, since delivery orders can be interdependent and cause critical path analysis and integrated master schedule obscuration if not managed and reported as a single program).

References and Resources

1. Department of Defense, December 2, 2008, Defense Acquisition Guidebook, Operations of the Defense Acquisition System, DoD Instruction Number 5000.02.

Additional References and Resources

Boehm, B., July 2000, Spiral Development: Experience, Principles, and Refinements, Spiral Development Workshop, February 9, 2000, CMU/SEI-2000-SR-008.

Dobbins, J., M. Kelley, and P. Sherlock, October 2009, DoD Acquisition Assurance for IT Systems, The MITRE Corporation.

Hansen, W.J., et al., July 2000, Spiral Development—Building the Culture, A Report on the CSE-SEI Workshop, February 2000, CMU/SEI-2000-SR-006.

Hansen, W.J., et al., May 2001, Spiral Development and Evolutionary Acquisition, The SEI-CSE Workshop, September 2000, CMU/SEI-2001-SR-005.

PROGRAM ACQUISITION STRATEGY FORMULATION

"Big-Bang" Acquisition

Definition: *An acquisition strategy is a high-level business and technical management approach designed to achieve program objectives within specified resource constraints. It is the framework for planning, organizing, staffing, controlling, and leading a program. The traditional strategy for system acquisition—also called "Big-Bang," "Grand Design," or "One-Shot"—involves a single pass through the organization's acquisition life cycle.*

Keywords: *acquisition, big bang, grand design, one pass, one shot*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the fundamental conditions and assumptions under which an acquisition program/project can be successfully pre-specified, planned, and controlled so that only one pass through the traditional systems engineering life cycle is necessary to deliver a system or capability. MITRE SEs are also expected to understand when a big-bang acquisition is not appropriate to a situation, explain the basis for their assessment, and recommend better alternatives.

Background

Max Wideman says, "What happens in a traditional acquisition process? Simply put, you figure out what you want, describe it in a Request for Proposal, solicit bids, pick a competent vendor with the lowest price and fastest delivery, enter into a standard legal contract, wait for the work to be finished, and come back when you can 'pick up the keys to the front door'[1]." This strategy was used extensively by the Department of Defense (DoD), as well as other government agencies and foreign governments, for the acquisition of major systems [2]. In the mid-1990s, it was replaced by evolutionary acquisition as the preferred strategy. (See the article "Evolutionary Acquisition.")

Increasingly, government departments and agencies are acquiring information technology (IT)-intensive systems and capabilities, and deploying rapidly fielded systems. Recent studies by the Defense Science Board (DSB) [3, 4] have concluded what many have suspected for years: "The deliberate process through which weapon systems and information technology are being acquired does not match the speed at which new IT capabilities are being introduced in today's information age." Thus big-bang acquisitions are expected to become less relevant and less frequently applied to future government capability developments.

Best Practices and Lessons Learned

The big-bang strategy is based on a number of assumptions (usually unstated), such as:

- The user and the acquisition organization can define and articulate all system requirements in the request for proposal.
- The critical technologies for the system remain static from the time the proposal is requested until the system is tested and delivered.
- This type of system has been built before, and the development contractor knows how to build (or acquire) and integrate the necessary subsystems and components.
- The interfaces with other systems remain static from the time the proposal is requested until the system is tested and delivered.
- The user's operational environment does not change from initial request through delivery of the product.
- The contractor will deliver the requested product without substantial interim review by the acquisition organization.
- The government's original cost estimate was accurate, product funding remains "protected" for the life of the contract, and management reserve is available to handle known unknowns.
- The acquisition organization can coordinate and integrate the acquisition with other interfacing systems and parallel development efforts.

In the acquisition of systems where these assumptions hold true, big-bang acquisition is the most efficient approach. However, when the program is canceled before final delivery, the customer does not receive expended funding to date and must pay for any termination liability that was specified in the contract. When these assumptions do not hold and the program has elected (or been directed) to use a big-bang acquisition strategy, the program will require more time and funding and will probably deliver less functionality, performance, or quantity than originally specified. The typical response to these shortcomings is to add more funding rather than cancel the program (usually causing a reverse incentive to the developing contractor).

Figure 1 illustrates the General Accountability Office's assessment of the F/A-22 program for using a big-bang acquisition strategy [5]. The F/A-22's advanced avionics, intelligence, and communications technologies were not available at the time

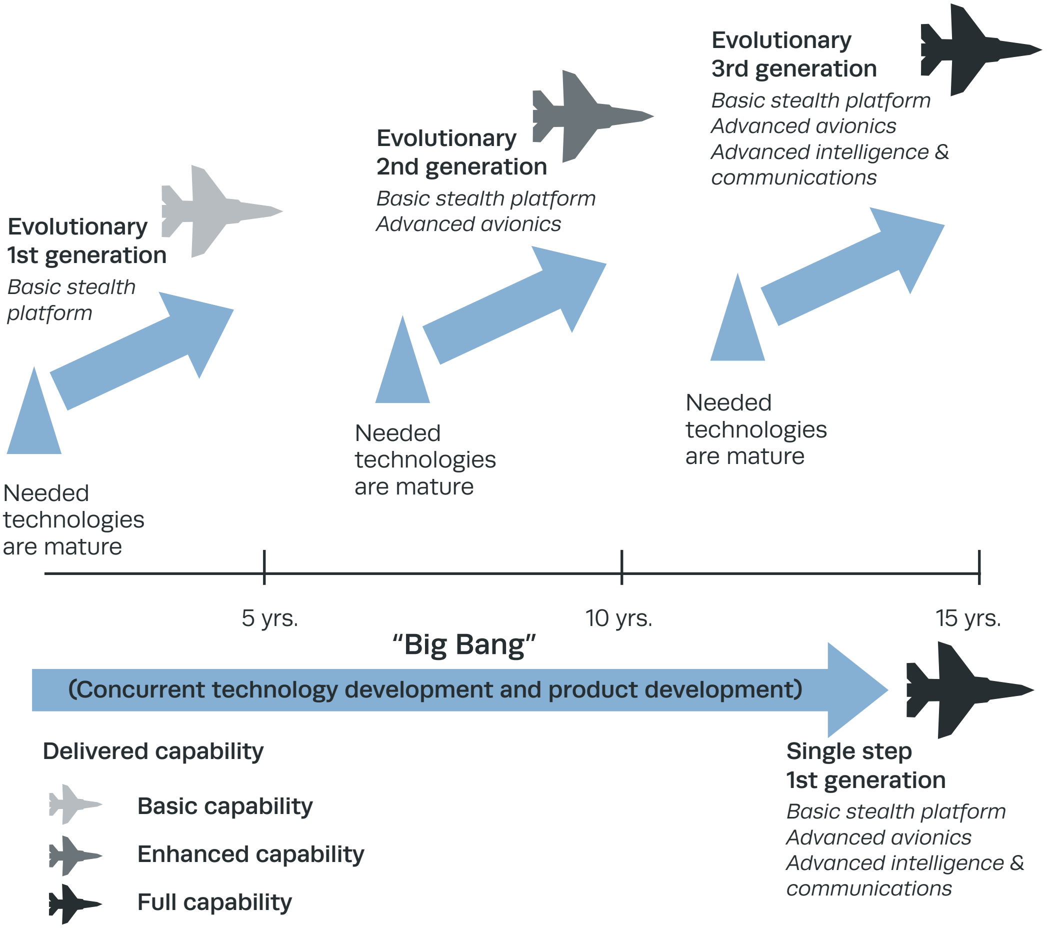


Figure 1. Comparison of "Big-Bang" and Evolutionary Acquisition

of initial contract award. Rather than developing the basic stealth platform with provisions for later technology insertion, a version of big-bang acquisition was used in which the entire aircraft delivery was delayed while the avionics, intelligence, and communications technologies matured.

Prior to the early or mid-1990s, big-bang acquisition was the normal approach for doing business in the DoD. With the exception of the SR-71 "Skunkworks" development, there have been few cases in the last four decades where a government big-bang development was completed fast enough for all of the assumptions to hold. Commercial aircraft and automobile firms have had better success with the big-bang strategy because of their shorter development cycles. When it became obvious that long acquisition times for major system acquisitions were causing the assumptions to fail, the department policy was changed to favor evolutionary acquisition. Evolutionary acquisition is an incremental approach to delivery of capability, providing for quicker initial (or partial) delivery of capability, while allowing future increments or spirals to address the balance of the requirements and accommodate changes. (See the article "Evolutionary Acquisition.")

The systems engineering implications of a big-bang acquisition are tied closely to the assumptions listed above. These assumptions have been learned through MITRE's experience with traditional big acquisitions and explain why many large programs fail despite good traditional systems engineering practice. The longer an acquisition program remains in the development phase, the more likely there will be changes to the requirements, environment or mission, or relevant technology requiring contract modifications and engineering change proposals, thereby lengthening the cycle for delivery and increasing the cost. For some programs, this becomes a vicious cycle (more changes beget more changes) and the development is never completed.

It is critical for the MITRE SE to point to the assumptions and stress their importance as indispensable to success when using big-bang as the acquisition strategy for a program. It is more likely that this strategy should never be chosen (based on historical lack of success) and an alternate strategy should be recommended.

References and Resources

1. Max's Project Management Wisdom website, "Progressive Acquisition and the RUP, Part I: Defining the Problem and Common Terminology."
2. Defense Systems Management College (DSMC), 1999, "Acquisition Strategy Guide," 4th Ed.
3. DSB Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology, March 2009, "Department of Defense Policies and Procedures for the Acquisition of Information Technology."
4. DSB Task Force on Future Perspectives, April 2009, "Creating a DoD Strategic Platform."
5. GAO Testimony, April 11, 2003, "Better Acquisition Outcomes Are Possible If DoD Can Apply Lessons From F/A-22 Program."

Contractor Evaluation

Definition: *Contractor evaluation is an activity to assess the contractor's technical and programmatic progress, approaches, and deliverables. The purpose of contractor evaluation is to provide insight into risks and the likelihood of meeting program and contractual requirements [1].*

Keywords: *contractor evaluation, contractor performance, milestone reviews*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) support contractor evaluations and milestone reviews, influence sponsor/customer decisions during those reviews, monitor the contractor's continued performance, and recommend changes based on their performance [1]. MITRE SEs are expected to apply strong domain and technical expertise and experience and perform with objectivity consistent with the FFRDC role.

Context

Contractor evaluation is a component of performance management. It is a process for making course corrections that uses performance information to adjust resources and activities to achieve an organization's end goals. The focus of performance management is on the future: What do you need to be able to do, and how can you do things better? Managing performance is about managing for results.

MITRE teams frequently are asked to lead and participate in contractor evaluation activities because the characteristics of federally funded research and development centers (FFRDCs), as chartered under Federal Acquisition Regulation (FAR) 35.017 [2], promote independence, objectivity, freedom from conflicts of interest, and technical expertise. These characteristics enable the MITRE team to provide findings and recommendations that might reduce risk to the government program and increase the probability of a favorable outcome.

This topic contains three articles. The article "Data-Driven Contractor Evaluations and Milestone Reviews" provides guidance to MITRE SEs who monitor, assess, and recommend improvements to a contractor's technical and programmatic approaches, work packages, prototypes, and deliverables before and during reviews. The other two articles—"Earned Value Management" and "Competitive Prototyping"—provide guidance and lessons learned on key specific techniques for monitoring contractor performance. Earned value management (EVM) integrates data on project scope, schedule, and cost to measure progress, and is required in many government programs. It gives the MITRE SE insight into potential program risks and can form the basis for making recommendations to mitigate those risks. Competitive prototyping (CP) is an approach in which two or more competing organizations develop prototypes during the early stages of a project. In a number of acquisition reform initiatives, the U.S. government has encouraged or required CP to be used as a tool to assess technology maturity and reduce program risk. The "Competitive Prototyping" article provides guidance on when to recommend competitive prototyping and offers best practices and lessons learned for monitoring and evaluating contractor competitive prototyping technical efforts. See related information in articles under the SEG's MITRE FFRDC Independent Assessments topic.

Best Practices and Lessons Learned

Maintain positive and professional relationships with all contractors. The contractor and government teams will be more receptive to MITRE findings and recommendations if they are developed and presented in a positive, professional atmosphere. In addition, occasionally MITRE SEs find themselves working with the same contractor or government team members on different projects. In that situation, the shared experience of a professional encounter can prove helpful in making progress with the new project.

Planning for roles and activities is essential. The government team, MITRE, and the contractor all have specific roles and responsibilities, and frequently MITRE is asked to lead the government team efforts in defining the technical components. MITRE also usually drafts the technical evaluation plan. The best practices and lessons learned sections in the articles under this topic provide guidance on both roles and responsibilities and planning.

References and Resources

Federal Acquisition Regulation (FAR), http://www.acquisition.gov/far/, accessed February 5, 2010.

The MITRE Corporation, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1.13E, pp. 39–40.

CONTRACTOR EVALUATION

Data-Driven Contractor Evaluations and Milestone Reviews

Definition: *Data-driven contractor evaluations and milestone reviews provide an objective assessment of contractor performance at technical milestone reviews. Technical reviews and the content to be addressed are typically prescribed by government agency or department mandates available to MITRE staff and other project members prior to the actual milestone.*

Keywords: *empirical data, independent technical assessments, metrics, milestone reviews, performance assessments, technical reviews*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to provide technical thought leadership and assessment throughout an entire government program life cycle. While ongoing insight is needed to quickly grasp and respond to program risks and opportunities, its importance peaks at event-driven milestones when key government decisions are made. At those times, MITRE SEs are expected to lead and participate in teams reviewing the contractor proposed technical approach. MITRE SEs analyze design review content against milestone entry and exit criteria to ensure that the contractor delivers quality products on time and within budget. They are expected to assess the contractor's technical and programmatic approaches, work packages, prototypes, and deliverables before and during reviews to identify issues and ensure that decision makers are provided with data-driven recommendations during technical and program milestone reviews [1].

Introduction

MITRE SEs can assume many roles at technical milestone reviews. Depending on the size and complexity of the program, many MITRE staff may be supporting the same technical review or, on some programs, only one or two. Staff typically perform as subject matter experts (SMEs) for specific technical areas (e.g., adequacy of requirements capture, maturity of the architecture) to be reviewed; they provide informal and formal assessments to the government sponsor. It is also not uncommon for MITRE to develop an overall assessment of the entire technical review. This assessment may include aggregating the input from MITRE staff and other program office contractor support. Whatever the scope, focus, or size of the MITRE review effort, the overall assessment must be based largely on empirical data, metrics, the trends they indicate, and demonstrated system performance. During reviews, MITRE staff need to be prepared, inquisitive, confident, technically competent, thorough, current with program progress, tactful in dealing with the contractor, and convincing in their overall assessments. Finally, MITRE's assessment of and recommendation on whether the technical review "passed" or "failed" can have a significant impact on whether the program meets its schedule or experiences long and costly delays.

Government Interest and Use

The government has myriad guidelines and mandates that define how systems should be acquired, developed, delivered, and sustained. In attempts to track the progress of a system development, the government has also defined a set of technical reviews to be conducted at various phases of development. Conducting these reviews successfully requires insight into contractor progress. Although it is a government responsibility to formally sign off on the final assessment of a technical review, MITRE is relied on heavily to provide convincing and credible technical evidence to support the assessment.

Independent, fact-based engineering analysis is essential to government program managers (PMs) in making their assessment of whether a program meets its technical review criteria.

Among the most critical times for MITRE to provide unbiased and technically substantiated assessments on a program is when supporting technical milestone reviews. We need to work with the contractor to ensure that the government PM is presented with empirical data and metrics that characterize system progress and performance as accurately as possible. That increases the likelihood that the government PM will make the right decision because it is based on objective data that supports the overall assessment.

It is important to ensure that technical recommendations are not influenced by the natural, collective desire of program stakeholders for the program to be viewed as a success and to move forward. Because of program pressures to succeed, technical assessments that indicate program problems may not be immediately embraced. In rare cases, it may be necessary to provide a formal, independent message of record to the PM documenting the technical assessment, the rationale for the perceived risk to the program (i.e., the likelihood of not meeting technical objectives, schedule, or cost and the impact), what may happen if the situation is not addressed, and recommended steps to mitigate the risk. The PM should be made aware of such a message and its contents personally before it is issued. While such a communication may not be welcomed in the short term, in the long run, it maintains the high standard that our customers expect of us.

Best Practices and Lessons Learned

Ensure consensus-based entry/exit criteria. The name, purpose, and general requirements of each technical review in standard acquisition processes are usually well defined in department or agency regulations [2]. What is often not done, but is essential for conducting a coordinated and successful technical review, is to ensure that the government team and contractor have documented formal entry and exit criteria, and that consensus has been reached on their content. If these do not exist, it is important to ensure that they are created and, if required, for MITRE staff to take responsibility for ensuring that they are defined. The entry/exit criteria should be tailored to meet the needs of each program. This is an area where MITRE can contribute—by emphasizing criteria (e.g., data, prototypes, metrics) that can be objectively assessed. Sample entry/exit criteria for many reviews are contained in the Mission Planning Technical Reviews [3].

Prepare, prepare, prepare. The backgrounds, skill sets, and experiences of the systems engineering team supporting the government at a technical review can vary widely. Depending on our role in the supported program, MITRE can and should instigate and lead government preparation meetings to ensure that entry/exit criteria are known, responsibilities of each SME are defined ahead of time, there is a pre-review artifacts/contract data requirements lists and government leadership attending have been "prepped" on strengths/weaknesses of the contractor and where they should weigh in. It is also beneficial to conduct technical review "dry runs" with the contractor prior to the review. At the same time, be sensitive to the demands that dry runs place on the contractor. Structure them to be less formal and intrusive while achieving the insight they provide. The benefits of these dry runs are:

- They require the contractor to prepare for the review earlier and reduce the possibility of them creating "just-in-time" charts for the major review that may have disappointing content from the government perspective. If the content falls short of expectations, there is time for them to correct it.

- They allow more people to attend a version of the review and have their questions answered, since meetings will be smaller. Though key PM and technical team members will attend both the dry run and final review, others are likely to attend only one.
- They allow a graceful way to reschedule the review if the contractor is not ready by dry run. This is especially important for programs that are under substantial scrutiny.

Divide and conquer. No one can know all aspects of a contractor's effort, regardless of how able the staff is, how long they have been on the program, or how technically competent they are. It may also happen that a program's systems engineering staff resources may be weighted in a particular discipline (e.g., software engineers, radar engineers, network specialists). Program technical reviews are all-encompassing. They must address user requirements, risk identification and mitigation, performance, architecture, security, testing, integration, and more. If staff resources are limited, it is advisable to assign SMEs who are strong in one discipline (e.g., software engineering) the secondary responsibility of another discipline (e.g., risk identification) at the technical review. This has the benefit of ensuring that all disciplines are covered at some level during the review and provides the opportunity to train staff in secondary systems engineering disciplines that broaden their skill set and help the government in the long run.

Gauge "ground truth" for yourself. Be aware of the true program progress well ahead of the review. Know the "real" workers responsible for day-to-day development, who may be different from those presenting progress reports at reviews. This will allow you to more accurately gauge progress. This requires advanced preparation, including meetings with programmers, attending contractor in-house peer reviews, reviewing development metrics, witnessing early prototype results, observing in-house testing, and spending time in the contractor's facility to know fact from fiction.

Assess when fresh. Recognize that technical reviews can be long, tedious, information packed, and physically and mentally draining events. As difficult as it may be, attempt to conduct a government team caucus at the end of each day to review what was accomplished and to gain preliminary team feedback. Meetings do not have to be long; a half hour can be sufficient. It is advantageous to gather the impressions of team members, since it can quickly confirm the review's formal presentations or uncover differences. Use the entry/exit criteria to voice what was "satisfactory" and what was not. Finally, when it is time to aggregate all input for the entire review, it is valuable to have the daily reviews to streamline the assembly of the formal assessment.

Use mostly data, part "gut feeling." Though it is desirable for the technical reviews to be civil, "just the facts" affairs, there may be times when exchanges become contentious and relationships between government and contractor representatives become strained. Personalities can get involved and accusations may be made, which are driven more by defensive instincts than impartial assessment of data. This is the time to make maximum use of objective data to assess contractor progress and solution development maturity, while refraining from over-reliance on anecdotal information and subjective assertions. Metrics and the trends they illuminate should be used as the basis for questions during the review and assessments after the review. Metrics to demonstrate software size, progress, and quality, should be assessed. (For software-intensive systems, it may be advisable to compare productivity/defect rates to other industries [4], other military systems [5], or CMMI maturity level standards [6].) Preliminary data to indicate system performance, reliability, and user satisfaction should be examined and challenged if necessary. Staffing metrics can be used to corroborate sufficiency of assigned resources. Testing metrics should be reviewed, as well. Don't ignore "gut feelings," but use them selectively. When the data says one thing and your intuition says another, intensify your efforts to obtain additional fact-based evidence to reconcile the disparity.

Search for independence. Regardless of how knowledgeable organic project staff is on all phases of your acquisition and the technologies responsible for the most prominent program risks, it is advisable to call on independent SMEs for selected technical reviews. In fact, Department of Defense (DoD) guidance for the development of systems engineering plans, as well as the Defense Acquisition Guide (DAG), call out the need for independent SMEs. This is excellent advice. For large, critical, and high-visibility programs undergoing oversight by their respective department or agency acquisition authority, conducting an Independent Technical Assessment (ITA) to assess the maturity of the program at a major technical review (e.g., PDR, CDR) can help develop objective evidence to inform the final assessment. It may also be advisable to include an SME from a large, respected technical organization on the ITA to provide advice in their areas of special expertise (e.g., Carnegie Mellon Software Engineering Institute [SEI] on Capability Maturity Model issues). It may be advantageous to use a qualified, senior-level MITRE technical SME to lead the ITA, as a way of bringing the corporation to bear. It is also advisable to include a senior manager from the prime contractor being reviewed, as long as this person is *not* in the direct management chain of the program leadership. This can open many doors with the prime contractor that may have seemed closed in the past. Recognize that bringing on independent SMEs for a review has a distinct cost (e.g., organic staff resources will need to bring SME members up to speed). However, judiciously done, it can be worthwhile.

References and Resources

1. MITRE Systems Engineering (SE) Competency Model, Ver. 1, September 1, 2007, p. 38.
2. Defense Acquisition Guidebook, Chapter 4.
3. 951st Electronic Systems Group, April 2007, "Mission Planning Technical Reviews."
4. Jones, C., April 2008, *Applied Software Measurement: Global Analysis of Productivity and Quality*, 3rd Ed., McGraw-Hill Osborne Media.

5. Reifer, J., July 2004, "Industry Software Cost, Quality and Productivity Benchmarks," *The DoD Software Tech News,* Vol. 7, Number 2.
6. Croxford, M., and R. Chapman, May 2005, "Correctness by Construction: A Manifesto for High-Integrity Software," *Crosstalk: The Journal of Defense Software Engineering.*

Additional References and Resources

Jones, C., May 2000, *Software Assessments, Benchmarks, and Best Practices*, Addison-Wesley Professional.

Neugent, B., August 2006, "How to Do Independent Program Assessments," The MITRE Corporation.

Definition: *Earned value management (EVM) is a technique for measuring project progress in an objective manner. It integrates technical scope, schedule, and cost for definitized contract work [1].*

Keywords: *contractor performance, earned value, earned value management system, performance-based earned value, performance measurement, planned value*

CONTRACTOR EVALUATION

Earned Value Management

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to sufficiently understand the principles and elements of EVM to monitor and assess contractor performance, and use its results as the basis for recommending program changes.

Background

Monitoring contractor performance consists of measuring and evaluating the contractor's progress to assess the likelihood of meeting program and contractual requirements for cost, schedule, and technical viability [2]. Performance measurement is part of performance management, a process for making course corrections to achieve an organization's goals by using performance information to adjust resources and activities. The focus of performance management is the future: What do you need to be able to do, and how can you do things better? Managing performance is about managing for results [3].

A widely used practice for monitoring contractor performance by managers and SEs is reviewing and assessing earned value management (EVM) results. In the 1990s, many U.S. government regulations were eliminated or streamlined. However, EVM not only survived the streamlining, but emerged as a financial analysis specialty that has since become a significant branch of project management and cost engineering for both government and industry. Today, more and more civilian agencies are requiring EVM in their contracts to better manage and ensure successful outcomes. In accordance with OMB Circular A-11, Part 7, agencies must use a performance-based acquisition management system, based on ANSI/EIA Standard 748, to measure achievement of the cost, schedule, and performance goals [4].

Basic Concepts of Earned Value Management

EVM is a technique used to track the progress and status of a project and forecast the likely future performance of the project. EVM integrates technical scope with the time-phased cost or budget required to complete the scope to facilitate integrated management of program planning and execution [1]. It can result in meeting the technical scope within cost and schedule parameters, reducing or eliminating schedule delays, and reducing or eliminating cost overruns [1]. The basic elements of the EV are depicted in Figure 1 [5]. Specifically, earned value (EV) consists of three dimensions: (1) The plan or budgeted cost of work scheduled (BCWS), (2) The performance or budgeted cost of work performed (BCWP), and (3) The cost of performance or actual cost of work performed (ACWP). These three data elements are used as the basis for computing and analyzing project performance.

Monitoring Contractor Performance Using EVM

For purposes of monitoring contractor performance, EVM is useful because it provides quantitative or earned value data that can be used to assess how well the contractor is performing. It also provides an early warning of performance problems.

Earned value data such as schedule and cost variances stem from a comparison of:

- The planned budget and the amount of budget earned for work accomplished
- The budget earned with the actual direct costs for the same work.

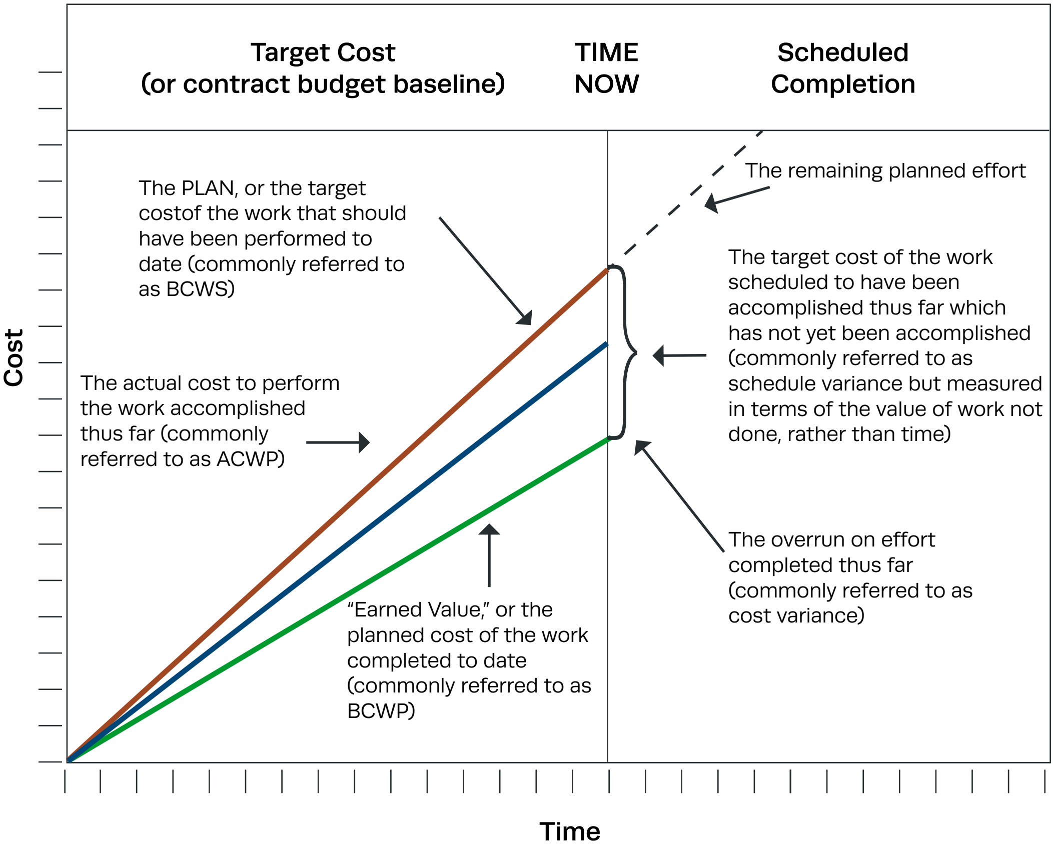

Target Cost
(or contract budget baseline)
TIME
NOW
Scheduled
Completion
Cost
Time
The remaining planned effort
The PLAN, or the target costof the work that should have been performed to date (commonly referred to as BCWS)
The target cost of the work scheduled to have been accomplished thus far which has not yet been accomplished (commonly referred to as schedule variance but measured in terms of the value of work not done, rather than time)
The actual cost to perform the work accomplished thus far (commonly referred to as ACWP)
The overrun on effort completed thus far (commonly referred to as cost variance)
"Earned Value," or the planned cost of the work completed to date (commonly referred to as BCWP)

Figure 1. Basic EVM Concept of Total Contract Performance

Schedule and cost indices provide information such as cost over/underruns, schedule delays/or ahead of schedule, etc. Other EV data elements that can be used to measure/monitor contractor performance are the schedule performance index (SPI) and the cost performance index (CPI). The SPI is defined as:

$$SPI = \frac{BCWP}{BCWS}$$

or the ratio of the BCWP over the BCWS. The SPI is a pure dimensionless quantity, insensitive to inflation, that measures schedule efficiency. A value above 1.00 indicates that the work performed to date is ahead of schedule. In other words, the SPI is an index measuring the efficiency of time utilization.

CPI is an index showing the efficiency of resource utilization and is defined as [6]:

$$CPI = \frac{BCWP}{ACWP}$$

or the ratio of BCWP to ACWP. CPI is an earned-value metric that measures cost efficiency where a value above 1.00 indicates that the work performed to date cost less than originally planned. A possible reason for CPI > 1 may be a high employee turnover rate. High turnover slows the pace of projects and the associated labor costs because there are gaps in the work stream due to employee departures and training issues.

The CPI and SPI are statistically and mathematically related because they share one variable in common: BCWP.

$$SPI = \frac{(ACWP)(CPI)}{BCWS}$$

Thus, if all of the variation in the data is dominated by BCWP, then the SPI and CPI will exhibit high correlation.

The SPI and CPI performance metrics compare baselines to actuals.

EVM Case Scenario

The best way to illustrate EVM is through an example [4]:

> Scenario: Building an aquarium budgeted at $5,500 by the end of November. At the end of November, spent $5,600 and only accomplished $4,900 worth of work.
>
> EV calculation: BCWS = $5,500, BCWP = $4,900, ACWP = $5,600; Cost Variance = -$700; Schedule Variance = -$600
>
> EV Analysis: In the month of November, spent $5,600 but only accomplished $4,900 worth of work; therefore there is a cost overrun of $700 as well as a delay in schedule of $600.

The preceding EV data elements can be used to compute the schedule and cost variance as follows:

Schedule variance = BCWP – BCWS

Cost variance = BCWP – ACWP

This example is depicted in Figure 2.

The next step for the MITRE SE would be to review these results and determine whether action is needed. Because EVM assesses the combined interaction of scope, schedule, and cost, an unfavorable EVM report tells the SE that if these three variables remain fixed and there is no change in how project performance is achieved, the project is at risk to achieve the three objectives. Remember the balloon analogy—if you push in one spot, another must give.

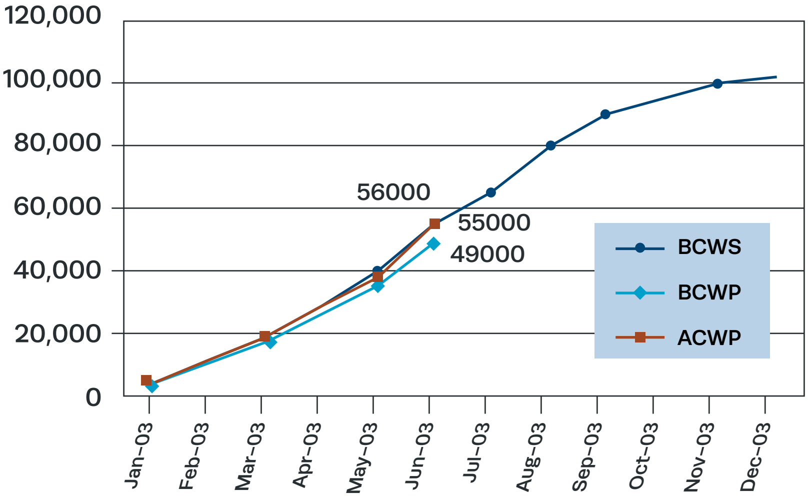


Figure 2. EVM Example

If the EVM results indicate action is needed, as it is in this example, the SE should review options to discuss with the government program manager. These options include:

- Accept the schedule delay and find the additional funding to cover the cost overrun and schedule delay.
- Recommend changing either the scope, schedule, or budget for the program or project. When this option is chosen, it is frequently for expediency. This is one of the reasons why we often see projects delivered on time and within budget with a reduction in the originally specified functionality (scope).
- Recommend changing an underlying condition that is causing the unfavorable EVM results. These include:
  - **Scope:** Requirements have increased beyond those manageable by the allocated schedule and cost. Suggest returning to the original scope or adjusting the cost and schedule accordingly.
  - **Schedule:** Consider how the schedule was determined. Was it by an engineering analysis and a work breakdown structure, or, as sometimes happens, was it determined by an imposed date? Suggest structuring a realistic schedule or adjusting the scope and cost accordingly.
  - **Cost:** Consider the productivity of the team and whether unrealistic assumptions were made in the original plans. Consider adding experienced, exceptionally capable staff. However, keep in mind that, in general, increasing staff significantly usually

will not erase lost productivity. Suggest adjusting the schedule and cost to accommodate actual productivity, or reduce scope to match the productivity.

Situations vary and any of these three approaches might be appropriate. For example, sometimes the solution must be implemented quickly, and expediency can override the option of digging into the underlying conditions of the unfavorable EVM results.

Value of EVM

A sound EVM implementation provides the following contractor performance data: [7]

- Relates time-phased budgets to specific contract tasks and/or statements of work
- Objectively measures work progress
- Properly relates cost, schedule, and technical accomplishment
- Allows for informed decision making and corrective action
- Is valid, timely, and can be audited
- Allows for statistical estimation of future costs
- Supplies managers at all levels with status information at the appropriate level
- Derives from the same EVM system used by the contractor to manage the contract.

The EV technique enhances the cost performance analysis of a project. Traditional cost analysis centers on the actual cost of the completed work. Therefore, much progress has been made to collect the actual costs through time charge and accounting systems that exist on practically all projects. What EV brings to the process is a measure of the amount of work that has been done in a unit of measure that is consistent and comparable with costs [8].

Best Practices and Lessons Learned

EVM does not measure the quality and technical maturity of the evolving work products. Other project mechanisms should be used to assess product quality and maturity as well as the appropriateness of the scope and conformance to requirements. For more information, see the articles “Data-Driven Contractor Evaluations and Milestone Reviews” and “Planning and Managing Independent Assessments.” Performance-based earned value (PBEV) is an enhancement to the EVMS standard for measuring technical performance and quality. It is based on standards and models for systems engineering, software engineering, and project management [9]. Ultimately, aside from cost and schedule, the PBEV adds the technical element into the mix.

EVM data is reliable and accurate if and only if the following occur [9]:

- The indicated quality of the evolving product is measured
- The right base measures of technical performance are selected
- Progress is objectively assessed.

Use EVM to mitigate risk by analyzing the results. For example, if targets are not being met, are they realistic? Which activities are making the greatest impact? What factors in the project or organizational culture contribute to results? Performance measures tell you what is happening, but they do not tell you why it is happening. See the article "How to Develop a Measurement Capability" and the Risk Management topic.

Contractor source selection certification considerations. EVM certification compliance may be contained in the Request for Proposal (RFP). However, not all RFPs require EVM compliance. Depending on what is written in the RFP regarding EVM, the contractor's EVMS can be evaluated by ensuring that its EVMS is American National Standards Institute/Electronic Industries Alliance (ANSI/EIA) 748 compliant. The ANSI/EIA 748 standard defines 32 criteria, which are intended to provide guidance, structure, and process direction for the successful operation of EVMS. These 32 criteria provide a basis for evaluating a contractor's EVMS. The best way to ensure that the contractor meets this standard is for the contractor to acquire an EVMS certification from the Defense Contract Management Agency or from a third-party vendor. In the absence of EVMS certification, a contractor's EVMS can be evaluated based on its EVMS plan that includes the 32 ANSI/EIA criteria. See the article "Acquisition Management Metrics."

Contractor proposal considerations. The contractor's proposal should demonstrate understanding of EVM by containing knowledge of industry standards and applying them specifically to the project. The EVMS should align with the integrated master schedule (IMS). The control account details should represent the work breakdown structure that drives the discrete deliverables that can be associated with cost and responsible resources. The EVM training listed on the IMS should specifically be tailored to the project. The IMS is the impetus to effective planning and scheduling. It contains planned events and milestones, accomplishments, exit criteria, and activities from contract award to the completion of the contract. It also allows for critical path analysis, forecasting, and a baseline plan. This scheduling aspect has to be linked to EVM in order for EVM analysis to be accurate and effective. See the article "Integrated Master Schedule (IMS)/Integrated Master Plan (IMP) Application."

EVM is not an effective tool for level-of-effort (LOE) activities. For non-schedule-based contracts (i.e., contracts composed primarily of LOE), EVM may not be effectively implemented due to a lack of measurement on work efforts that cannot be segmented. With that said, the LOE method can be used for measuring EV; however, it is primarily reserved for tasks that are time-related rather than task-oriented (i.e., tasks that have no measurable output). The LOE method has no schedule variance; therefore, it should not be used for any tasks with a schedule that might slip or be variable [1].

Evaluate the EVMS effectiveness. Ultimately the measurement of a successful EVMS depends on the customer's ability to use the information generated from the system and to evaluate the contractor's ability to manage the project [9]. The EVMS can be costly to maintain, so it is important to periodically consider its effectiveness and whether changes should be made. Recent U.S. government policy initiatives have been introduced to better facilitate customer insight into contractor performance. The reduced reporting threshold is down to $20 million, and policy revisions related to the integrated baseline review will have a significant impact on the administration and performance of contracts [10].

References and Resources

1. Society of Cost Estimating and Analysis (SCEA), 2002, "Tracking cost and schedule performance on project," *Earned Value Management Systems (EVMS)*.
2. United States Government Accountability Office, May 2005, Performance Measurement and Evaluation: Definitions and Relationships, GAO-05-739SP.
3. The MITRE Institute, "Introduction to Enterprise Business Strategy," *Performance Management*.
4. Executive Office of the President, Office of Management and Budget, August 2009, Circular No. A-11 Preparation, Submission, and Execution of the Budget.
5. The MITRE Corporation, October 11, 2005, Earned Value Management: A Quick Review for Busy Executives, Slide 17.
6. Tutorials Point website, Earned Value Management: Cost Variance.
7. Ernst, K. D., October 2006, *Earned Value Management Implementation Guide*, p. 2.
8. Wilkens, T. T., April 1, 1999, *Earned Value, Clear and Simple*, p. 4, Los Angeles County Transit Authority.
9. Solomon, P., August 2005, "Performance-Based Earned Value," *CrossTalk: The Journal of Defense Software Engineering*.
10. Johnson, C., April 2006, "Implementing an ANSI/EIA-748-Compliant Earned Value Management System," *Contract Management*.

Additional References and Resources

"Earned Value Management," Wikipedia, accessed October 15, 2009.

McKinlay, M., April 2006, "Why Not Implement EVM?" International Cost Engineering Council, *ICEC Cost Management Journal*.

Seigle, J., May 19, 2006, Earned Value Management Demystified Version 2.0, The MITRE Corporation.

The MITRE Corporation, "Contractor Evaluation," MITRE Systems Engineering Competency Model.

USAID, Earned Value Management.

Definition: *Competitive Prototyping is an approach in which two or more competing teams (organizations) develop prototypes during the early stages of a project (acquisition or procurement phase). The competing prototypes are compared, and ultimately one is chosen that best addresses the issue(s), problem(s), or challenge(s).*

Keywords: *competitive prototype program, competitive prototype tests, competitive prototyping strategy, competitive teams*

CONTRACTOR EVALUATION

Competitive Prototyping

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of competitive prototyping (CP) in the acquisition process, where it occurs in systems development, and the benefits and risks of employing it. MITRE SEs are also expected to understand and recommend when CP is appropriate to a situation. They are expected to develop and recommend technical requirements for CP efforts as well as strategies and processes that encourage and facilitate active participation of end users and other stakeholders in the CP process. They are expected to monitor and evaluate contractor CP technical efforts and the acquisition program's overall CP processes and recommend changes when warranted.

Background

Prototyping is a practice in which an early sample or model of a system, capability, or process is built to answer specific questions about, give insight into, or reduce uncertainty or risk in many diverse areas. This includes exploring alternative concepts, technology maturity assessments, requirements discovery or refinement, design alternative assessments, and performance or suitability issues. It is part of the SE's toolkit of techniques for managing uncertainty and complexity and mitigating their effects.

The exact form and focus of prototyping is driven by where it is used in the acquisition management system life cycle and the nature of the problem the prototype is intended to address. Prototyping may be used immediately after a decision to pursue a material solution to meet an operational need (see Figure 1). In this situation, prototypes are used to examine alternative concepts as part of the analysis of alternatives leading to a preferred solution concept. Prototyping to explore and evaluate the feasibility of high-level conceptual designs may be performed early in technology development as part of government activities to assess and increase technology maturity, discover or refine requirements, or develop a preliminary design. A prototype may even be developed into a reference implementation and provided to a commercial contractor for production. Competitive prototyping may serve any of the just-cited purposes, but it typically involves two or more competing teams, usually from commercial contractors, and is often used as a factor in source selection evaluations leading to a formal acquisition program start and contract award. This application of competitive prototyping is depicted in Figure 1.

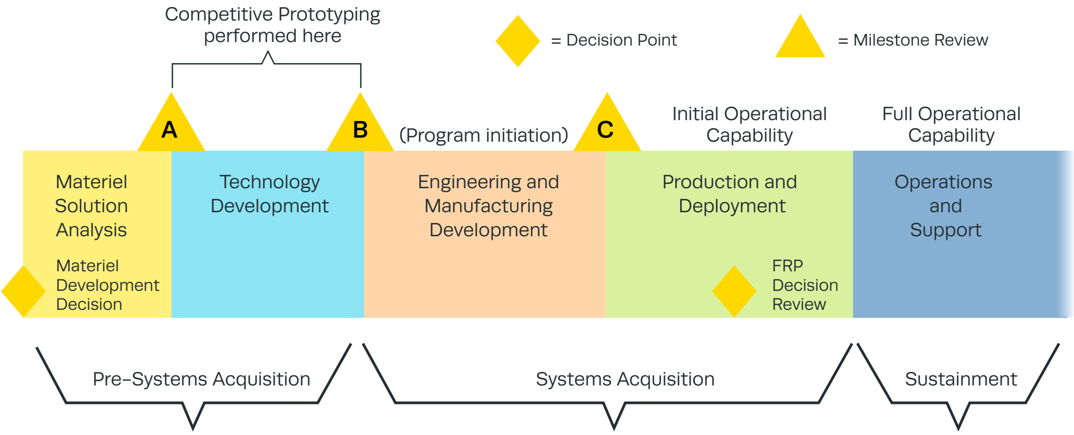

Competitive Prototyping performed here
= Decision Point
= Milestone Review
A
B
(Program initiation)
C
Initial Operational Capability
Full Operational Capability
Materiel Solution Analysis
Materiel Development Decision
Technology Development
Engineering and Manufacturing Development
Production and Deployment
FRP Decision Review
Operations and Support
Pre-Systems Acquisition
Systems Acquisition
Sustainment

Figure 1. The Defense Acquisition Management System

Historically much of competitive prototyping focused on building tangible prototypes such as weapons, aircraft, and automobiles. The earliest documented modern use of competitive prototyping dates back to just after the War of 1812, when the United States Army became interested in a breech-loading rifle [1]. Numerous contractors submitted actual hardware examples for the Army's evaluation. After a hardware design was accepted, the contractor was given an order for a limited production of these rifles.

The U.S. aircraft industry used CP extensively throughout the 20th century. Some of the early prototypes were developed by Chanute, the Wright Brothers, Curtiss, and Sikorsky and, more recently, by General Dynamics, Boeing, and McDonnell Douglas. Similarly, the U.S. auto industry uses competitive teams to design concept cars of the future.

The employment of CP in software-intensive system developments is a relatively recent phenomenon.

Government Interest and Use

In several acquisition reform initiatives, the U.S. government encouraged or required competitive prototyping as a tool to assess technology maturity and reduce program risk. Although mentioned less frequently as a primary reason, competitive prototyping can also illuminate undiscovered or uncertain requirements before engineering and manufacturing development.

The Office of Management and Budget has identified competitive prototyping as a risk mitigation tool to be used in procurement and cites five advantages for its use [2]:

- Proves concepts are sound.
- Allows efficient and effective communication (among operational users, procurement agency, and commercial contractors) to identify the best fit between agency (operational user) needs and marketplace capabilities.
- Provides for competition during the development effort.
- Where appropriate, ensures development remains constrained.
- Facilitates firm fixed-price contracting for production.

The strongest and most recent government support for competitive prototyping comes from the Department of Defense (DoD), which now requires all programs to formulate acquisition strategies and funding that provide for two or more competing teams to produce prototypes through Milestone B [3], as shown in Figure 1. High-level DoD acquisition officials from previous administrations have also endorsed this required use of competitive prototyping in government acquisitions [4].

Reasons noted in the DoD decision to require competitive prototyping are to enable government and industry teams to discover and solve technical issues before engineering and manufacturing development (EMD), so that EMD can focus on producing detailed manufacturing designs, not on solving myriad technical issues. Other anticipated advantages include:

reduced technical risk, validated design, validated cost estimates, insight into contractor manufacturing processes, refined requirements, and reduced time to fielding. But note that not all of these advantages can be expected to automatically accrue from all instances of competitive prototyping. For example, it is unlikely that substantial insight into contractor manufacturing processes to be used later for full-scale production will result from a prototyping effort unless considerable funding is devoted to tooling up for the prototype activity.

Best Practices and Lessons Learned [4, 5, 6]

When size (and skill) matters. Acquisition program offices that employ CPs successfully tend to require a larger contingent of government systems engineers with greater than average technical competence. Although this may appear counter-intuitive, remember that CPs offer advantages to programs that use them, but they must be skillfully planned, monitored, and managed by the government team.

Right-sizing CP requirements. CP is an investment that buys information to reduce uncertainty and risk. But CP adds up-front costs to a program right at a time when funding may be scarce and support for the program is often weak. A CP may run into opposition from the least expected stakeholders—staunch advocates of a program who believe that it must be pushed at great speed to fill capability gaps. To navigate these external forces on CP efforts, the program CP requirements must be right-sized. They must focus on areas that have substantial risk or offer a high reward-risk ratio, whatever and wherever those areas may be—high-level capabilities/levels-of-service, low-level detailed requirements at the subsystem level, or issues in between. It is also important to make sure that likely performance bottlenecks are identified in the prototype process that are measurable and measured as part of prototype testing.

Make sure your CP learns from antecedent activities. One focus of recent government acquisition reform initiatives is on the importance of early systems engineering. Some departments and agencies are strongly recommending or mandating prototyping in advance of technology development, during materiel solution analysis (see Figure 1). Results or lessons learned from these very early prototypes should be used to shape and inform CP activities.

Have your CP do double duty. The primary purpose of CP is to illuminate and eliminate technology maturity risks. But don't lose sight of the fact that a CP can give important insight into other risk areas such as contractor manufacturing processes (if the CP is resourced appropriately) and undiscovered operational user requirements. Look for and collect information on all issues and areas that a CP can illuminate, especially important downstream systems engineering activities and assessments for which CP information can form the basis of refined government assessments of system design or estimates of program cost.

Ensure persistent, active engagement of all stakeholders. CP is not a competition between two gladiators in an arena slugging it out until one gives in, at which time everyone else in the coliseum looks up and applauds the winner. CP efforts must be structured to encourage active participation of end users and other stakeholders throughout the CP life cycle. To facilitate that involvement, CP efforts should emphasize frequent demonstrations of progress and evidence that a prototype can scale. Ideally CPs should be developed iteratively or in an evolutionary fashion and be able to demonstrate interim operational capabilities. Active operational user stakeholder engagement is particularly critical to CPs intended to address requirements discovery and refinement.

Remember those without "skin in the game." Important stakeholders in the eventual outcome of a program, like certification and accreditation authorities, are frequently forgotten during CP. Identify and bring these stakeholders into CP planning early so they can advise on "non-starters" and be engaged through the entire process.

Commercial competitors are stakeholders, too. CPs are viewed as investments by commercial industry. To attract the best commercial competitors for your program, CP goals must be clearly defined and any basis for industry investment (e.g., internal research and development) must be convincing. In particular, the production potential of the contract must be visible and attractive to would-be competitors.

Don't stop competition too quickly. Make sure there is sufficient information to make informed decisions before terminating a competition. This is a form of the wisdom, "measure twice, cut once." The Joint Strike Fighter program began with a competition involving prototypes built by Boeing and Lockheed Martin. During the CP phase, it appeared that both prototypes had been extensively flown before the government chose the Lockheed variant, but rising costs and schedule slips of the F-35 now suggest that competition may have been closed too quickly.

Beware the Potemkin Village. Ensure each competitor is presenting an actual prototype and not simply performing a demonstration. A demonstration can be scripted to perform a few things well when conducted by a knowledgeable person. On the other hand, a prototype should be able to be operated without a script and by a domain expert with very little experience with the prototype.

Keep your eyes on the prize. Acquisitions using CPs can overemphasize the operator functional features of the prototypes at the expense of other critical, less obvious requirements. If competitors hit the mark on "operator functionality" or "user interface look and feel" more or less the same, there may be a risk of eliminating the better contractor for production. Carefully evaluate the potential risks of a prototype becoming the actual product. Prototypes often do not have robust architectures, standard designs, a full set of requirements, or complete documentation. These weaknesses may become deployment risks such as lack of maintainability, scalability, or reproducibility. Also, consider how big a challenge it would be for the functionality or look and feel of the other contractor's prototype to be modified and approved, as well as the contractor's ability and willingness to do so.

Retain competitors' core skills. CPs involve down-selects, but the intellectual capital and experience built up by "losers" need not and should not be lost to the program or the government. During the evaluation interval between each phase of a CP down-select, the government should continue to fund all competitors at a level of effort sufficient to retain the core engineering skills of the competing teams. Not doing so risks losing key members of competitors' teams and can weaken the overall government acquisition effort. One reference [5] suggests that investments in down-selected bidders can be capitalized on by structuring the ensuing acquisition so that unsuccessful bidders are offered "consolation prizes," such as independent verification and validation contracts. If this tack works at all, it is more likely to be attractive to small contractors or others attempting to break into the business area represented by the system being acquired.

References and Resources

1. Patnode, C.A. Jr., LTC, February 1973, Problems of Managing Competitive Prototype Programs Study Report. Defense Systems Management School (Program Management Course Student Study Program), PMC 73-2, p. 3.
2. Office of Management and Budget, June 2006, "Competitive Prototyping," section II.3.3 of Capital Programming Guide, V 2.0, Supplement to OMB Circular A-11, Part 7.
3. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, September 19, 2007, Memorandum on Prototyping and Competition, Washington, DC: Pentagon.
4. DuPont, D. G., February 2008, "Proactive Prototypes," *Scientific American*.
5. Boehm, B., and D. Ingold, July 2008, Initial Competitive Prototyping Survey Results, University of Southern California Center for Systems and Software Engineering. Presented at July 2008 OUSD/AT&L/SSE—USC/CSSE Workshop on Integrating Systems and Software Engineering with the Incremental Commitment Model.
6. Ireland, B., July 2008, Competitive Prototyping: Industry Roundtable, Presented at July 2008 OUSD/AT&L/SSE—USC/CSSE Workshop on Integrating Systems and Software Engineering with the Incremental Commitment Model.

Risk Management

Definition: *Risk is an event that, if it occurs, adversely affects the ability of a project to achieve its outcome objectives [1]. Risk management is the process of identifying risk, assessing risk, and taking steps to reduce risk to an acceptable level [2].*

Keywords: *opportunity, risk, risk analysis, risk management, uncertainty, uncertainty analysis*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) working on engineering systems are expected to propose, influence, and often design the risk management approach that enables risk-informed trade-offs and decisions to be made throughout a system's evolution. They are expected to identify, analyze, and prioritize risks based on impact, probabilities, dependencies, timeframes, and unknowns. They are expected to prepare and monitor risk mitigation plans and strategies, conduct reviews, and elevate important risks [3].

Context

Risk management lies at the intersection of project functions performed by the systems engineer and the project manager [3]. Historically risk management focused more on management elements such as schedule and cost, and less on technical risks for well-defined or smaller projects. However, larger and more complex projects and environments have increased the uncertainty for the technical aspects of many projects. To increase the likelihood of successful project and program outcomes, the SE and project manager must be actively involved in all aspects of risk management.

A substantial body of knowledge has developed around risk management. In general, risk management includes development of a risk management approach and plan, identification of components of the risk management process, and guidance on activities, effective practices, and tools for executing each component. One characterization of the risk management process is shown in Figure 1 [1].

- **Step 1. Risk Identification:** This is the critical first step of the risk management process. Its objective is the early and continuous identification of risks, including those within and external to the engineering system project.

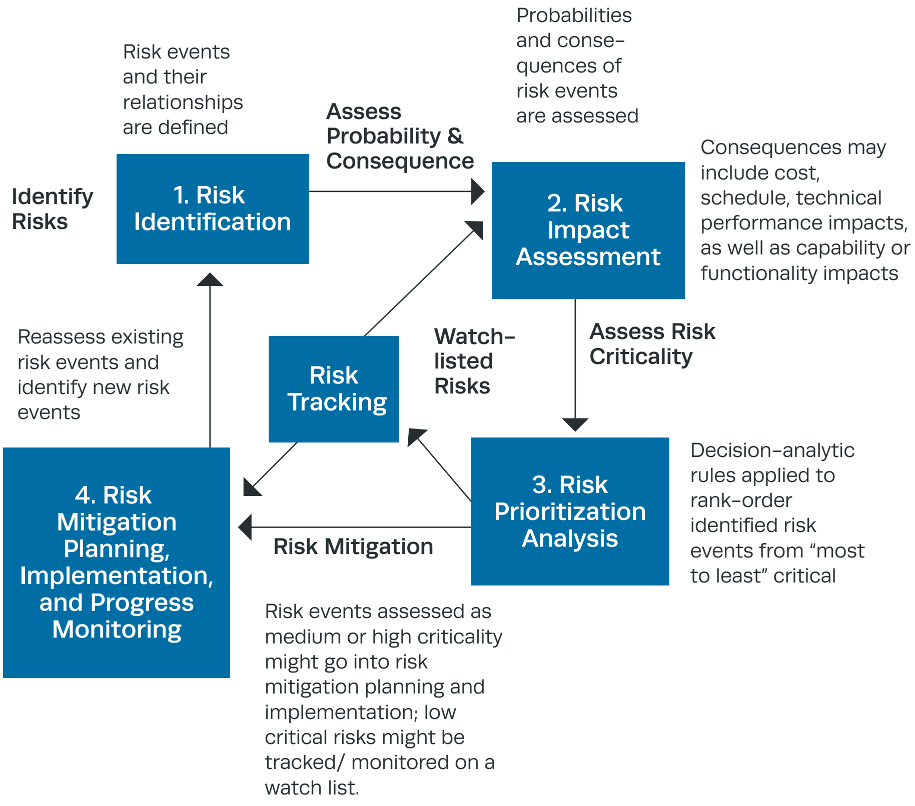

Risk events and their relationships are defined
Identify Risks
1. Risk Identification
Assess Probability & Consequence
Probabilities and conse-quences of risk events are assessed
2. Risk Impact Assessment
Consequences may include cost, schedule, technical performance impacts, as well as capability or functionality impacts
Reassess existing risk events and identify new risk events
Risk Tracking
Watch-listed Risks
Assess Risk Criticality
4. Risk Mitigation Planning, Implementation, and Progress Monitoring
Risk Mitigation
3. Risk Prioritization Analysis
Decision-analytic rules applied to rank-order identified risk events from "most to least" critical
Risk events assessed as medium or high criticality might go into risk mitigation planning and implementation; low critical risks might be tracked/ monitored on a watch list.

Figure 1. Fundamental Steps of Risk Management

- **Step 2. Risk Impact or Consequence Assessment:** An assessment is made of the impact each risk event could have on the engineering system project. Typically this includes how the event could impact cost, schedule, or technical performance objectives. Impacts are not limited to only these criteria. Additional criteria such as political or economic consequences may also require consideration. In addition, an assessment is made of the probability (chance) each risk event will occur.
- **Step 3. Risk Prioritization:** The overall set of identified risk events, their impact assessments, and their occurrence probabilities are "processed" to derive a most critical to least critical rank-order of identified risks. A major purpose for prioritizing risks is to form a basis for allocating critical resources.
- **Step 4. Risk Mitigation Planning:** This step involves the development of mitigation plans designed to manage, eliminate, or reduce risk to an acceptable level. Once a plan is implemented, it is continually monitored to assess its efficacy with the intent to revise the course of action, if needed.

Two other steps are involved in executing risk management: developing the approach and plan, and selecting the risk management tools. The risk management approach determines the processes, techniques, tools, and team roles and responsibilities for a specific project. The risk management plan describes how risk management will be structured and performed on the project [4]. Risk management tools support the implementation and execution of program risk management in systems engineering programs. In selecting the appropriate tools, the project team considers factors such as program complexity and available resources.

These six steps are discussed in the five articles under the SEG's Risk Management topic.

Risk Management Principles

MITRE SEs supporting government customers in risk management activities have observed the following elements common to the Department of Defense (DoD) and civilian environments.

Risk Management Is Fundamental

An event is uncertain if there is indefiniteness about its outcome [1]. Risk management acknowledges the concept of uncertainty, which includes risks (unfavorable outcomes) and opportunities (favorable outcomes). Risk management is a formal and disciplined practice for addressing risk. In many ways, it is indistinguishable from program management. It includes identifying risks, assessing their probabilities and consequences, developing management strategies, and monitoring their state to maintain situational awareness of changes in potential threats.

Every Project Involves Risk

Every project is a temporary endeavor undertaken to provide a unique result [3]; it is an undertaking that has not been done before. Therefore, all projects involve some level of risk, even if similar projects have been completed successfully.

Risk and Opportunity Must Be Balanced

Risk and opportunity management deal with uncertainty that is present throughout the systems' life cycle. The objective is to achieve a proper balance between them, while recognizing one is not the complement of the other.

Typically more risk and opportunity is involved in decisions that are made early in the project life cycle because those decisions have a more significant impact on project scope, cost, and schedule than those made later in the life cycle.

Risk Is Present in Complicated Relationships

Risk affects all aspects of engineering a system, and can be present in complicated relationships among project goals. A system may be intended for technical accomplishments near the limits of engineering or the maturity of technology, leading to technical risks. System development may be deployed too early to meet an imminent threat, thus resulting in schedule risks. All systems have funding challenges, which lead to cost risks. Risk can be introduced by external threats, due to changing social, political, or economic landscapes.

References and Resources

1. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.
2. Stoneburner, G., A. Goguen, and A. Feringa, July 2002, Risk Management Guide for Information Technology System, National Institute of Standards and Technology, Special Publication 800-30, p. 1.
3. The MITRE Corporation, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1, pp. 10, 40–41.
4. Project Management Institute, *A Guide to the Project Management Body of Knowledge, (PMBOK Guide),* Fourth Edition, ANSI/PMI 99-001-2008, pp. 273–312.

Additional References and Resources

IEEE Standard for Software Life Cycle Processes - Risk Management, IEEE Std. 1540-2001.

International Council on Systems Engineering (INCOSE), January 2010, *INCOSE Systems Engineering Handbook*, Version 3.2, INCOSE-TP-2003-002-03.2, p. 213–225.

International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC Guide 73, ISO/IEC Guide 73, Risk Management Vocabulary Guidelines.

Kerzner, H., 2003, *Project Management,* Eighth Ed., John Wiley & Sons, Inc., pp. 651–710.

Kossiakoff, A. and W. N. Sweet, 2003, *Systems Engineering Principles and Practice,* John Wiley and Sons, Inc., pp. 98–106.

MITRE Systems Engineering Practice Office, Risk Management Toolkit.

Moore, J. W., 2006, *The Road Map to Software Engineering, A Standards-Based Guide*, IEEE Computer Society, pp. 171–172.

Mulcahy, R., 2003, *Risk Management: Tricks of the Trade for Project Managers,* RMC Publications.

OMB Circular A-11 E-300, June 2008, "Part 7: Planning, Budgeting, Acquisition and Management of Capital Assets."

Software Engineering Institute CMMI, "Risk and Opportunity Management."

Thayer, R. H., and M. Dorfman (eds.), 2005, *Software Engineering Volume 2: The Supporting Processes,* Third Ed., IEEE Computer Society.

The Institute of Risk Management, "The Risk Management Standard."

Woodward, D., and K. Buck, July 2007, "Office of Management and Budget (OMB) Uncertainty and Risk Assessment Requirements: A Preliminary MITRE Study (MP #070137)."

Definition: *Risk management is the process of identifying risk, assessing risk, and taking steps to reduce risk to an acceptable level [1]. The risk management approach determines the processes, techniques, tools, and team roles and responsibilities for a specific project. The risk management plan describes how risk management will be structured and performed on the project [2].*

Keywords: *risk management, risk management approach, risk management plan, risk management process*

RISK MANAGEMENT

Risk Management Approach and Plan

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) working on government programs propose and influence, and often design, the risk management approach. They prepare and monitor risk mitigation plans and strategies for the government project or program office, and they review risk management plans prepared by government contractors [3].

Background

As a management process, risk management is used to identify and avoid the potential cost, schedule, and performance/technical risks to a system, take a proactive and structured approach to manage negative outcomes, respond to them if they occur, and identify potential opportunities that may be hidden in the situation [4]. The risk management approach and plan operationalize these management goals.

Because no two projects are exactly alike, the risk management approach and plan should be tailored to the scope and complexity of individual projects. Other considerations include the roles, responsibilities, and size of the project team, the risk management processes required or recommended by the government organization, and the risk management tools available to the project.

Risk occurs across the spectrum of government and its various enterprises, systems of systems, and individual systems. At the system level, the risk focus typically centers on development. Risk exists in operations, requirements, design, development, integration, testing, training, fielding, etc. (see the SEG's SE Life-Cycle Building Blocks section). For systems of systems, the dependency risks rise to the top. Working consistency across the system of systems, synchronizing capability development and fielding, considering whether to interface, interoperate, or integrate, and the risks associated with these paths all come to the forefront in the system-of-systems environment. At the enterprise level, governance and complexity risks become more prominent. Governance risk of different guidance across the enterprise for the benefit of the enterprise will trickle down into the system of systems and individual systems, resulting in potentially unanticipated demands and perhaps suboptimal solutions at the low level that may be beneficial at the enterprise level. Dealing with the unknowns increases, and the risks associated with these—techniques in the SEG's Enterprise Engineering section such as loose couplings, federated architectures, and portfolio management—can help the MITRE SE alleviate these risks.

Risk Management in System–Level Programs

System-level risk management is predominantly the responsibility of the team working to provide capabilities for a particular development effort. Within a system-level risk area, the primary responsibility falls to the system program manager and SE for working risk management, and the developers and integrators for helping identify and create approaches to reduce risk. In addition, a key responsibility is with the user community's decision maker on when to accept residual risk after it and its consequences have been identified. The articles in the Risk Management topic area provide guidance for identifying risk ("Risk Identification"), mitigating risks at the system level with options like control, transfer, and watch ("Risk Mitigation Planning, Implementation, and Progress Monitoring"), and a program risk assessment scale and matrix ("Risk Impact Assessment and Prioritization"). These guidelines, together with MITRE SEs using tools such as those identified in the "Risk Management Tools" article, will help the program team deal with risk management and provide realism to the development and implementation of capabilities for the users.

Risk Management in System-of-Systems Programs

Today, the body of literature on engineering risk management is largely aimed at addressing traditional engineering system projects—those systems designed and engineered against a set of well-defined user requirements, specifications, and technical standards. In contrast, little exists on how risk management principles apply to a system whose functionality and performance is governed by the interaction of a set of highly interconnected, yet independent, cooperating systems. Such systems may be referred to as systems of systems.

A system of systems can be thought of as a set or arrangement of systems that are related or interconnected to provide a given capability that, otherwise, would not be possible. The loss of any part of the supporting systems degrades or, in some cases, eliminates the performance or capabilities of the whole.

What makes risk management in the engineering of systems of systems more challenging than managing risk in a traditional systems engineering project? The basic risk management process steps are the same. The challenge comes from implementing and managing the process steps across a large-scale, complex system of systems—one whose subordinate systems, managers, and stakeholders may be geographically dispersed, organizationally distributed, and may not have fully intersecting user needs.

How does the delivery of capability over time affect how risks are managed in a system of systems? The difficulty is in aligning or mapping identified risks to capabilities planned to be delivered within a specified build by a specified time. Here, it is critically important that assessments are made as a function of which capabilities are affected, when these effects occur, and their impacts on users and stakeholders.

Lack of clearly defined system boundaries, management lines of responsibility, and accountability further challenge the management of risk in the engineering of systems of systems. User and stakeholder acceptance of risk management, and their participation in the process, is essential for success.

Given the above, a program needs to establish an environment where the reporting of risks and their potential consequences is encouraged and rewarded. Without this, there will be an incomplete picture of risk. Risks that threaten the successful engineering of a system of systems may become evident only when it is too late to effectively manage or mitigate them.

Frequently a system of systems is planned and engineered to deliver capabilities through a series of evolutionary builds. Risks can originate from different sources and threaten the system of systems at different times during their evolution. These risks and their sources should be mapped to the capabilities they potentially affect, according to their planned delivery date. Assessments should be made of each risk's potential impacts to planned capabilities, and whether they have collateral effects on dependent capabilities or technologies.

In most cases, the overall system-of-systems risk is not just a linear "roll-up" of its subordinate system-level risks. Rather, it is a combination of specific lower level individual system risks that, when put together, have the potential to adversely impact the system of systems in ways that do not equate to a simple roll-up of the system-level risks. The result is that some risks will be important to the individual systems and be managed at that level, while others will warrant the attention of system-of-systems engineering and management.

Risk Management in Enterprise Engineering Programs

Risk management of enterprise systems poses an even greater challenge than risk management in systems-of-systems programs.

Enterprise environments (e.g., the Internet) offer users ubiquitous, cross-boundary access to wide varieties of services, applications, and information repositories. Enterprise systems engineering is an emerging discipline. It encompasses and extends "traditional" systems engineering to create and evolve "webs" of systems and systems of systems that operate in a network-centric way to deliver capabilities via services, data, and applications through an interconnected network of information and communications technologies. This is an environment in which systems engineering is at its "water's edge."

In an enterprise, risk management is viewed as the integration of people, processes, and tools that together ensure the early and continuous identification and resolution of enterprise risks. The goal is to provide decision makers an enterprise-wide understanding of risks, their potential consequences, interdependencies, and rippling effects within and beyond enterprise "boundaries." Ultimately risk management aims to establish and maintain a holistic view of risks across the enterprise, so capabilities and performance objectives are achieved via risk-informed resource and investment decisions.

Today we are in the early stage of understanding how systems engineering, engineering management, and social science methods weave together to create systems that "live" and "evolve" in enterprise environments.

Requirements for Getting Risk Management Started

- Senior leadership commitment and participation is required.
- Stakeholder commitment and participation is required.
- Risk management is made a program-wide priority and "enforced" as such throughout the program's life cycle.

- Technical and program management disciplines are represented and engaged. Both program management and engineering specialties need to be communicating risk information and progress toward mitigation. Program management needs to identify contracting, funding concerns, SEs need to engage across the team and identify risks, costs, and potential ramifications if the risk were to occur, as well as mitigation plans (actions to reduce the risk, and cost/resources needed to execute successfully).
- Risk management is integrated into the program's business processes and systems engineering plans. Examples include risk status included in management meetings and/or program reviews, risk mitigation plan actions tracked in schedules, and cost estimates reflective of risk exposure.

The Risk Management Plan

The Risk Management Plan describes a process, such as the fundamental steps shown in Figure 1 in the preceding "Risk Management" topic article, that are intended to enable the engineering of a system that is accomplished within cost, delivered on time, and meets user needs.

Best Practices and Lessons Learned

Twenty-one "musts." In supporting both Department of Defense (DoD) and civilian agency projects and programs, MITRE SEs have found the following minimum conditions needed to initiate and continuously execute risk management successfully. With these, the program increases its chance of identifying risks early so the goals and objectives are achieved [5].

1. Risk management must be a priority for leadership and throughout the program's management levels. Maintain leadership priority and open communication. Teams will not identify risks if they do not perceive an open environment to share risk information (messenger not shot) or management priority on wanting to know risk information (requested at program reviews and meetings), or if they do not feel the information will be used to support management decisions (lip service, information not informative, team members will not waste their time if the information is not used).
2. Risk management must never be delegated to staff that lack authority.
3. A formal and repeatable risk management process must be present—one that is balanced in complexity and data needs, such that meaningful and actionable insights are produced with minimum burden.
4. The management culture must encourage and reward identifying risk by staff at all levels of program contribution.

5. Program leadership must have the ability to regularly and quickly engage subject matter experts.
6. Risk management must be formally integrated into program management.
7. Participants must be trained in the program's specific risk management practices and procedures.
8. A risk management plan must be written with its practices and procedures consistent with process training.
9. Risk management execution must be shared among all stakeholders.
10. Risks must be identified, assessed, and reviewed continuously—not just prior to major reviews.
11. Risk considerations must be a central focus of program reviews.
12. Risk management working groups and review boards must be rescheduled when conflicts arise with other program needs.
13. Risk mitigation plans must be developed, success criteria defined, and their implementation monitored relative to achieving success criteria outcomes.
14. Risks must be assigned only to staff with authority to implement mitigation actions and obligate resources.
15. Risk management must never be outsourced.
16. Risks that extend beyond traditional impact dimensions of cost, schedule, and technical performance must be considered (e.g., programmatic, enterprise, cross-program/cross-portfolio, and social, political, economic impacts).
17. Technology maturity and its future readiness must be understood.
18. The adaptability of a program's technology to change in operational environments must be understood.
19. Risks must be written clearly using the Condition-If-Then protocol.
20. The nature and needs of the program must drive the design of the risk management process with which a risk management tool/database conforms.
21. A risk management tool/database must be maintained with current risk status information; preferably, employ a tool/database that rapidly produces "dashboard-like" status reports for management.

It is important for MITRE SEs as well as project and program leaders to keep these minimum conditions in mind, with each taking action appropriate for their roles.

Get top-level buy-in. MITRE SEs can help gain senior leadership support for risk management by highlighting some of the engineering as well as programmatic risks. MITRE SEs should prepare assessments of the impact that risks could manifest and back them by facts and data (e.g., increased schedule due to more development, increased costs, increased user training for unique, technology-edge capabilities, and potential of risk that capabilities will not be used because they do not interoperate with legacy systems). MITRE SEs can highlight the various risk areas, present the pros and cons of alternative courses of mitigation actions (and their impacts), and help the decision makers determine the actual discriminators and residual impact of taking one action or another. In addition to data-driven technical assessments, success in getting top-level buy-in requires consideration of political, organizational/operational, and economic factors as seen through the eyes of the senior leadership [6].

Get stakeholder trust. Gain the trust of stakeholders by clearly basing risk reduction or acceptance recommendations on getting mission capabilities to users.

Leverage your peers. Someone at MITRE generally knows a lot about every risk management topic imaginable. This includes technical, operational, and programmatic dimensions of risks and mitigations. Bringing the company to bear is more than a slogan—it is a technique to use, as risks are determined, particularly in system-of-systems and enterprise programs. In all likelihood, MITRE is working other parts of these large problems.

Think horizontal. Emphasize cross-program or cross-organization participation in risk identification, assessment, and management. Cross-team coordination and communication can be particularly useful in risk management. All "-ilities" (e.g., information assurance, security, logistics, software) should be represented in the risk reviews. Communication of risk information helps illuminate risks that have impact across organizations and amplifies the benefits of mitigations that are shared.

Stay savvy in risk management processes and tools. Become the knowledgeable advisor in available risk management processes and tools. Many government organizations have program management offices that have defined risk management processes, templates, and tools. These should be used as a starting point to develop the specific approach and plan for an individual project or program. Make sure the government sponsors or customers have the current information about the risk management approach and plan required by their organizations, and assist them in complying with it. Assist the sponsors or customers in determining the minimum set of activities for their particular program that will produce an effective risk management approach and plan.

References and Resources

1. National Institute of Standards and Technology, July 2002, *Risk Management Guide for Information Technology System,* Special Publication 800-30, p. 1.
2. Project Management Institute, *A Guide to the Project Management Body of Knowledge, (PMBOK Guide),* Fourth Edition, ANSI/PMI 99-001-2008, pp. 273–312.
3. The MITRE Institute, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model, Ver. 1," pp. 10, 41–42.

4. International Council on Systems Engineering (INCOSE), January 2010, INCOSE Systems Engineering Handbook, Ver. 3.2, INCOSE-TP-2003-002-03.2, pp. 213–225.

5. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.

6. Neugent, B. "Persuasion," The MITRE Corporation.

Additional References and Resources

Kossiakoff, A. and W. N. Sweet, 2003*, Systems Engineering Principles and Practice,* John Wiley and Sons, Inc., pp. 98–106.

MITRE SEPO Risk Management Toolkit.

Definition: *Risk identification is the process of determining risks that could potentially prevent the program, enterprise, or investment from achieving its objectives. It includes documenting and communicating the concern.*

Keywords: *risk, risk identification, risk management*

RISK MANAGEMENT

Risk Identification

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) working on government programs are expected to identify risks that could impact the project and program. They are expected to write and review risk statements that are clear, unambiguous, and supported by evidence [1].

Background

Risk identification is the critical first step of the risk management process depicted in Figure 1 in the "Risk Management" topic article.

The objective of risk identification is the early and continuous identification of events that, if they occur, will have negative impacts on the project's ability to achieve performance or capability outcome goals. They may come from within the project or from external sources.

There are multiple types of risk assessments, including program risk assessments, risk assessments to support an investment decision, analysis of alternatives, and assessments of operational or cost uncertainty. Risk identification needs to match the type of assessment required to support risk-informed decision making. For an acquisition program, the first step is to identify the program goals and objectives, thus fostering a common understanding across the team of what is needed for program success. This gives context and bounds the scope by which risks are identified and assessed.

Identifying Risks in the Systems Engineering Program

There are multiple sources of risk. For risk identification, the project team should review the program scope, cost estimates, schedule (to include evaluation of the critical path), technical maturity, key performance parameters, performance challenges, stakeholder expectations vs. current plan, external and internal dependencies, implementation challenges, integration, interoperability, supportability, supply chain vulnerabilities, ability to handle threats, cost deviations, test event expectations, safety, security, and more. In addition, historical data from similar projects, stakeholder interviews, and risk lists provide valuable insight into areas for consideration of risk.

Risk identification is an iterative process. As the program progresses, more information will be gained about the program (e.g., specific design), and the risk statement will be adjusted to reflect the current understanding. New risks will be identified as the project progresses through the life cycle.

Best Practices and Lessons Learned

Operational risk. Understand the operational nature of the capabilities you are supporting and the risk to the end users, their missions, and their operations of the capabilities. Understanding of the operational need/mission (see the SEG's Concept Development topic) will help you appreciate the gravity of risks and the impact they could have to the end users. This is a critical part of risk analysis—realizing the real-world impact that can occur if a risk arises during operational use. Typically operational users are willing to accept some level of risk if they are able to accomplish their mission (e.g., mission assurance), but you need to help users to understand the risks they are accepting and to assess the options, balances, and alternatives available.

Technical maturity. Work with and leverage industry and academia to understand the technologies being considered for an effort and the likely transition of the technology over time. One approach is to work with vendors under a non-disclosure agreement to understand the capabilities and where they are going, so that the risk can be assessed.

Non-developmental items (NDI). NDI includes commercial-off-the-shelf and government-off-the-shelf items. To manage risk, consider the viability of the NDI provider. Does the provider have market share? Does the provider have appropriate longevity compared to its competitors? How does the provider address capability problems and release fixes, etc.? What is the user base for the particular NDI? Can the provider demonstrate the NDI, preferably in a setting similar to that of your customer? Can the government use the NDI to create a prototype? All of these factors will help assess the risk of the viability of the NDI and the provider. Seek answers to these questions from other MITRE staff that have worked the area or have used the NDI being assessed.

Acquisition drivers. Emphasize critical capability enablers, particularly those that have limited alternatives. Evaluate and determine the primary drivers to an acquisition and emphasize their associated risk in formulating risk mitigation recommendations. If a particular aspect of a capability is not critical to its success, its risk should be assessed, but it need not be the primary focus of risk management. For example, if there is risk to a proposed user interface, but the marketplace has numerous alternatives, the success of the proposed approach is probably less critical to overall success of the capability. On the other hand, an information security feature is likely to be critical. If only one alternative approach satisfies the need, emphasis should be placed on it. Determine the primary success drivers by evaluating needs and designs, and determining the alternatives that exist. Is a unique solution on the critical path to success? Make sure critical path analyses include the entire systems engineering cycle and not just development (i.e., system development, per se, may be a "piece of cake," but fielding in an active operational situation may be a major risk).

Use capability evolution to manage risk. If particular requirements are driving implementation of capabilities that are high risk due to unique development, edge-of-the-envelope performance needs, etc., the requirements should be discussed with the users for their criticality. It may be that the need could be postponed, and the development community should assess when it might be satisfied in the future. Help users and developers gauge how much risk (and schedule and cost impact) a particular capability should assume against the requirements to receive less risky capabilities sooner. In developing your recommendations, consider technical feasibility and knowledge of related implementation successes and failures to assess the risk of implementing now instead of in the future. In deferring capabilities, take care not to fall into the trap of postponing ultimate failure by trading near-term easy successes for a future of multiple high-risk requirements that may be essential to overall success.

Key performance parameters (KPPs). Work closely with the users to establish KPPs. Overall risk of program cancelation can be centered on failure to meet KPPs. Work with the users to ensure the parameters are responsive to mission needs and technically feasible. The parameters should not be so lenient that they can easily be met, but not meet the mission need; nor should they be so stringent that they cannot be met without an extensive effort or pushing technology—either of which can put a program at risk. Seek results of past operations, experiments, performance assessments, and industry implementations to help determine performance feasibility.

External and internal dependencies. Having an enterprise perspective can help acquirers, program managers, developers, integrators, and users appreciate risk from dependencies of a development effort. With the emergence of service-oriented approaches, a program will become more dependent on the availability and operation of services provided by others that they intend to use in their program's development efforts. Work with the developers of services to ensure quality services are being created, and thought has been put into the maintenance and evolution of those services. Work with the development program staff to assess the services that are available, their quality, and the risk that a program has in using and relying on the service. Likewise, there is a risk associated with creating the service and not using services from another enterprise effort. Help determine the risks and potential benefits of creating a service internal to the development with possibly a transition to the enterprise service at some future time.

Integration and interoperability (I&I). I&I will almost always be a major risk factor. They are forms of dependencies in which the value of integrating or interoperating has been judged to override their inherent risks. Techniques such as enterprise federation architecting, composable capabilities on demand, and design patterns can help the government plan and execute a route to navigate I&I risks. See the SEG's Enterprise Engineering section for articles on techniques for addressing I&I associated risks.

Information security. Information security is a risk in nearly every development. Some of this is due to the uniqueness of government needs and requirements in this area. Some of this is because of the inherent difficulties in countering cyber attacks. Creating defensive capabilities to cover the spectrum of attacks is challenging and risky. Help the government develop resiliency approaches (e.g., contingency plans, backup/recovery). Enabling information sharing across systems in coalition operations with international partners presents technical challenges and policy issues that translate into development risk. The information security issues associated with supply chain management are so broad and complex that even maintaining rudimentary awareness of the threats is a tremendous challenge.

Skill level. The skill or experience level of the developers, integrators, government, and other stakeholders can lead to risks. Be on the lookout for insufficient skills and reach across the corporation to fill any gaps. In doing so, help educate government team members at the same time you are bringing corporate skills and experience to bear.

Cost risks. Programs will typically create a government's cost estimate that considers risk. As you develop and refine the program's technical and other risks, the associated cost estimates should evolve, as well. Cost estimation is not a one-time activity.

Historical information as a guide to risk identification. Historical information from similar government programs can provide valuable insight into future risks. Seek out information about operational challenges and risks in various operation lessons learned, after-action reports, exercise summaries, and experimentation results. Customers often have repositories of these to access. Government leaders normally will communicate their strategic needs and challenges. Understand and factor these into your assessment of the most important capabilities needed by your customer and as a basis for risk assessments.

Historical data to help assess risk is frequently available from the past performance assessments and lessons learned of acquisition programs and contractors. In many cases, MITRE staff will assist the government in preparing performance information for a particular acquisition. The AF has a Past Performance Evaluation Guide (PPEG) that identifies the type of information to capture that can be used for future government source selections [3]. This repository of information can help provide background information of previous challenges and where they might arise again—both for the particular type of development activity as well as with the particular contractors.

Numerous technical assessments for vendor products can be accessed to determine the risk and viability of various products. One MITRE repository of evaluations of tools is the Analyst's Toolshed [4] that contains guidance on and experience with analytical tools. Using resources like these and seeking others who have tried products and techniques in prototypes and experiments can help assess the risks for a particular effort.

How to write a risk—a best practice [2]. A best practice protocol for writing a risk statement is the *Condition-If-Then* construct. This protocol applies to risk management processes designed for almost any environment. It is a recognition that a risk, by its nature, is probabilistic and one that, if it occurs, has unwanted consequences.

What is the *Condition-If-Then* construct?

- The *Condition* reflects what is known today. It is the root cause of the identified risk event. Thus the *Condition* is an event that has occurred, is presently occurring, or will occur with certainty. Risk events are future events that may occur because of the *Condition* present.
- The *If* is the risk event associated with the *Condition* present. It is critically important to recognize the *If* and the *Condition* as a dual. When examined jointly, there may be ways to directly intervene or remedy the risk event's underlying root (*Condition*) such that the consequences from this event, if it occurs, no longer threaten the project. The *If* is the probabilistic portion of the risk statement.

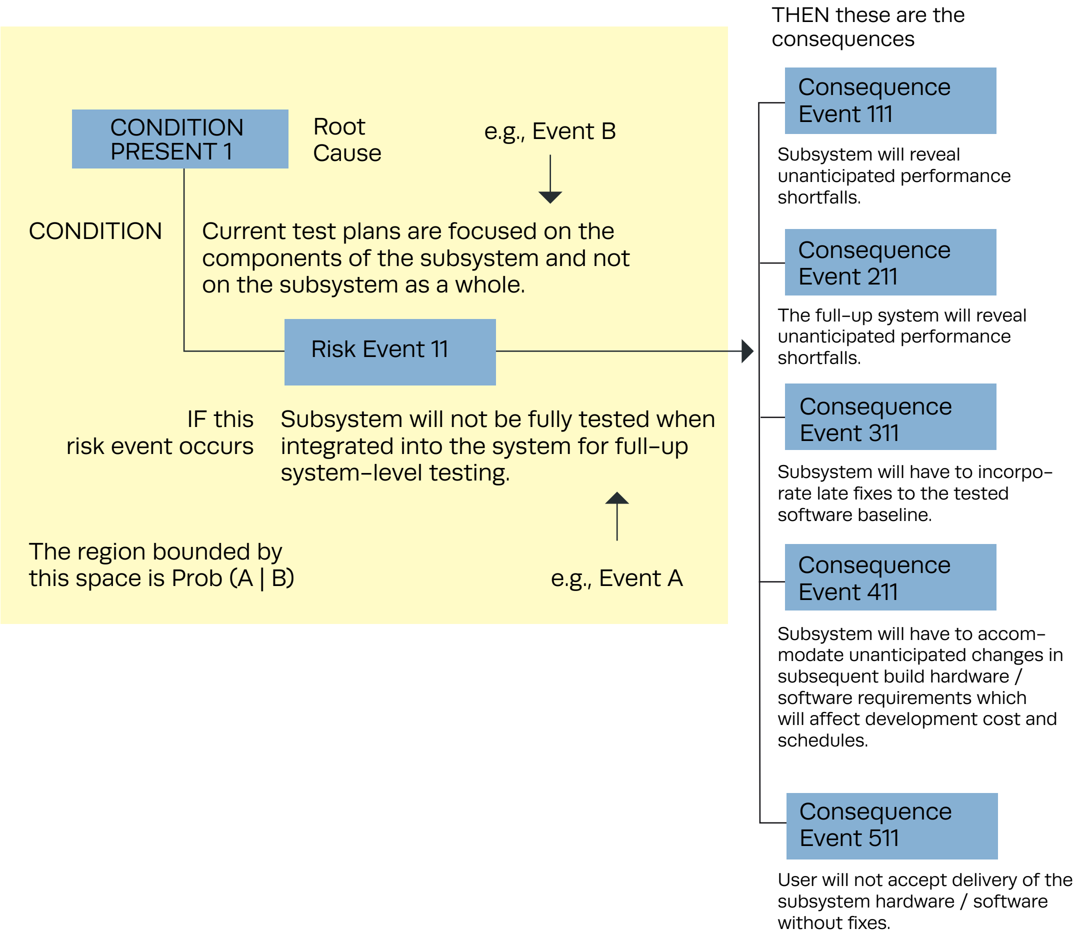

THEN these are the consequences
CONDITION PRESENT 1
Root Cause
e.g., Event B
CONDITION
Current test plans are focused on the components of the subsystem and not on the subsystem as a whole.
Risk Event 11
IF this risk event occurs
Subsystem will not be fully tested when integrated into the system for full-up system-level testing.
The region bounded by this space is Prob (A | B)
e.g., Event A
Consequence Event 111
Subsystem will reveal unanticipated performance shortfalls.
Consequence Event 211
The full-up system will reveal unanticipated performance shortfalls.
Consequence Event 311
Subsystem will have to incorporate late fixes to the tested software baseline.
Consequence Event 411
Subsystem will have to accommodate unanticipated changes in subsequent build hardware / software requirements which will affect development cost and schedules.
Consequence Event 511
User will not accept delivery of the subsystem hardware / software without fixes.

Figure 1. Writing a Risk—The *Condition-If-Then* Best Practice

- The *Then* is the consequence, or set of consequences, that will impact the engineering system project if the risk event occurs.

An example of a *Condition-If-Then* construct is illustrated in Figure 1.

Encourage teams to identify risks. The culture in some government projects and programs discourages the identification of risks. This may arise because the risk management activities of tracking, monitoring, and mitigating the risks are seen as burdensome and unhelpful. In this situation, it can be useful to talk to the teams about the benefits of identifying risks and the inability to manage it all in your heads (e.g., determine priority, who needs to be involved, mitigation actions).

Assist the government teams in executing a process that balances management investment with value to the outcomes of the project. In general, a good balance is being achieved when the project scope, schedule, and cost targets are being met or successfully mitigated by action plans, and the project team believes risk management activities provide value to the project. Cross-team representation is a must; risks should not be identified by an individual, or strictly by the systems engineering team (review sources of risk above).

Consider organizational and environmental factors. Organizational, cultural, political, and other environmental factors, such as stakeholder support or organizational priorities, can pose as much or more risk to a project than technical factors alone. These risks should be identified and actively mitigated throughout the life of the project. Mitigation activities could include monitoring legislative mandates or emergency changes that might affect the program or project mission, organizational changes that could affect user requirements or capability usefulness, or changes in political support that could affect funding. In each case, consider the risk to the program and identify action options for discussion with stakeholders. For more information, see the article" Risk Mitigation Planning, Implementation, and Progress Monitoring."

Include stakeholders in risk identification. Projects and programs usually have multiple stakeholders that bring various dimensions of risk to the outcomes. They include operators, who might be overwhelmed with new systems; users, who might not be properly trained or have fears for their jobs; supervisors, who might not support a new capability because it appears to diminish their authority; and policy makers, who are concerned with legislative approval and cost. In addition, it is important to include all stakeholders, such as certification and accreditation authorities who, if inadvertently overlooked, can pose major risks later in the program. Stakeholders may be keenly aware of various environmental factors, such as pending legislation or political program support that can pose risks to a project that are unknown to the government or MITRE project team. Include stakeholders in the risk identification process to help surface these risks.

Write clear risk statements. Using the *Condition-If-Then* format helps communicate and evaluate a risk statement and develop a mitigation strategy. The root cause is the underlying *Condition* that has introduced the risk (e.g., a design approach might be the cause), the If reflects the probability (e.g., probability assessment that the If portion of the risk statement were to occur), and the *Then* communicates the impact to the program (e.g., increased resources to support testing, additional schedule, and concern to meet performance). The mitigation strategy is almost always better when based on a clearly articulated risk statement.

Expect risk statement modifications as the risk assessment and mitigation strategy is developed. It is common to have risk statements refined once the team evaluates the impact. When assessing and documenting the potential risk impact (cost, schedule, technical, or timeframe), the understanding and statement of the risk might change. For example, when assessing a risk impact of software schedule slip, the risk statement might be refined to include the need-by date, and/or further clarification of impact (e.g. if the XYZ software is not delivered by March 2015, then there will not be sufficient time to test the interface exchanges prior to Limited User Test).

Do not include the mitigation statement in the risk statement. Be careful not to fall into the trap of having the mitigation statement introduced into the risk statement. A risk is an uncertainty with potential negative impact. Some jump quickly to the conclusion of mitigation of the risk and, instead of identifying the risk that should be mitigated (with mitigation options identified), they identify the risk as a suboptimal design approach. For example, a risk statement might be: If the contractor does not use XYZ for test, then the test will fail. The concern is really test sufficiency. If the contractor does not conduct the test with measurable results for analysis, then the program may not pass limited user test. Use of XYZ may be a mitigation option to reduce the test sufficiency risk.

Do not jump to a mitigation strategy before assessing the risk probability and impact. A risk may be refined or changed given further analysis, which might affect what the most efficient/desired mitigation may be. Engineers often jump to the solution; it is best to move to the next step discussed in the "Risk Impact Assessment and Prioritization" article to decompose and understand the problem first. Ultimately this will lead to a strategy that is closely aligned with the concern.

References and Resources

1. The MITRE Corproation, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1, pp. 10, 40–41.
2. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.
3. U.S. Air Force, January 2008, *Air Force Past Performance Evaluation Guide (PPEG)*, IG5315.305(a).
4. "Analysis Toolshed," MITREpedia, accessed March 2, 2010.

Additional References and Resources

MITRE E520 Risk Analysis and Management Technical Team checklists, Risk Checks, Risk Analysis and Management Documents.

Project Management Institute, *A Guide to the Project Management Body of Knowledge*, (PMBOK Guide), Fourth Edition, ANSI/PMI 99-001-2008, pp. 273–312.

"Standard Process/Steps of Process, Step 2: Identify Risks and Hazards," MITRE SEPO Risk Management Toolkit.

Definition: *Risk impact assessment is the process of assessing the probabilities and consequences of risk events if they are realized. The results of this assessment are then used to prioritize risks to establish a most-to-least-critical importance ranking. Ranking risks in terms of their criticality or importance provides insights to the project's management on where resources may be needed to manage or mitigate the realization of high probability/high consequence risk events.*

Keywords: *risk, risk impact assessment, risk management, risk prioritization*

RISK MANAGEMENT

Risk Impact Assessment and Prioritization

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) working on government programs are expected to analyze risks with respect to impact, probability, dependencies, and timeframes and to prioritize risks to facilitate decision making by the sponsor or customers [1].

Background

Risk impact assessment and prioritization are the second and third steps of the process depicted in Figure 1 in the Risk Management topic article [2].

Risk Impact Assessment in the Systems Engineering Program

In this step, the impact each risk event could have on the project is assessed. Typically this assessment considers how the event could impact cost, schedule, or technical performance objectives. Impacts are not limited to these criteria, however; political or economic consequences may also need to be considered. The probability (chance) each risk event will occur is also assessed. This often involves the use of subjective probability assessment techniques, particularly if circumstances preclude a direct evaluation of the probability by objective methods (i.e., engineering analysis, modeling, and simulation). Chapters 2 and 4 of Garvey [2] discuss the topic of subjective probability assessments, as well as criteria for assessing a risk event's impact or consequence to a project.

As part of the risk assessment, risk dependencies, interdependencies, and the timeframe of the potential impact (near-, mid-, or far-term) need to be identified. The MITRE-developed RiskNav® tool is an example of a tool that can help perform this assessment. For additional details, see the article "Risk Management Tools."

When assessing risk, it is important to match the assessment impact to the decision framework. For program management, risks are typically assessed against cost, schedule, and technical performance targets. Some programs may also include oversight and compliance, or political impacts. Garvey [2] provides an extensive set of rating scales for making these multicriteria assessments, as well as ways to combine them into an overall measure of impact or consequence. These scales provide a consistent basis for determining risk impact levels across cost, schedule, performance, and other criteria considered important to the project. In addition, the Risk Matrix tool can help evaluate these risks to particular programs (see the article "Risk Management Tools"). Performing POET (Political, Operational, Economic, Technical) and/or SWOT (Strengths, Weaknesses, Opportunities, and Threats) assessments can help determine the drivers of the risks. For more details on these analyses, see the Tools to Enable a Comprehensive Viewpoint topic in the SEG's Enterprise Engineering section.

For some programs or projects, the impacts of risk on enterprise or organizational goals and objectives are more meaningful to the managing organization. Risks are assessed against the potential negative impact on enterprise goals. Using risk management tools for the enterprise and its components can help with the consistency of risk determination. This consistency is similar to the scale example shown below, except that the assessment would be done at the enterprise level. Depending on the criticality of a component to enterprise success (e.g., risk of using commercial communications to support a military operation and the impact of the enterprise to mission success, versus risk of using commercial communications for peacetime transportation of military equipment), the risks may be viewed differently at the enterprise level even when the solution sets are the same or similar.

One way management plans for engineering an enterprise is to create capability portfolios of technology programs and initiatives that, when synchronized, will deliver time-phased capabilities that advance enterprise goals and mission outcomes. A capability portfolio is a time-dynamic organizing construct to deliver capabilities across specified epochs; a capability can be defined as the ability to achieve an effect to a standard under specified conditions using multiple combinations of means and ways to perform a set of tasks [2]. With the introduction of capability management, defining the impact of risk on functional or capability objectives may provide valuable insights into what capability is at risk, and which risks could potentially significantly impact the ability to achieve a capability and/or impact multiple capability areas.

In portfolio management, a set of investments is administered based on an overall goal(s), timing, tolerance for risk, cost/price interdependencies, a budget, and changes in the relevant environment over time. These factors are generally applicable to the government acquisition environment (see the article "Portfolio Management" in the SEG's Enterprise Engineering section). For portfolio risk assessment, investment decision, or analysis of alternatives tasks, using categories of risk area scales may be the most appropriate way to ensure each alternative or option has considered all areas of risk. Risk areas may include advocacy, funding, resources, schedule and cost estimate confidence, technical maturity, ability to meet technical performance, operational deployability, integration and interoperability, and complexity. Scales are determined for each risk area, and each alternative is assessed against all categories. Risk assessment may also include operational consideration of threat and vulnerability. For cost-risk analysis, the determination of uncertainty bounds is the risk assessment.

When determining the appropriate risk assessment approach, it is important to consider the information need. As a Probability of Occurrence example, the sample Program Risk Management Assessment Scale in Table 1 and the Investment Risk Assessment Scale in Table 2 are from MITRE's systems engineering work with government sponsors or clients.

Risk Prioritization in the Systems Engineering Program

In the risk prioritization step, the overall set of identified risk events, their impact assessments, and their probabilities of occurrences are "processed" to derive a most-to-least-critical rank-order of identified risks. A major purpose of prioritizing risks is to form a basis for allocating resources.

Multiple qualitative and quantitative techniques have been developed for risk impact assessment and prioritization. Qualitative techniques include analysis of probability and impact, developing a probability and impact matrix, risk categorization, risk frequency

Table 1. Sample Program Risk Management Assessment Scale

1.00	Issue:	1	Certain to occur
0.95–0.99	High:	> 0.95 < 1	Extremely sure to occur
0.85–0.95	High:	> 0.85 <= 0.95	Almost sure to occur
0.75–0.85	High:	> 0.75 <=0.85	Very likely to occur
0.65–0.75	High:	> 0.65 <=0.75	Likely to occur
0.55–0.65	Medium:	> 0.55 <=0.65	Somewhat greater than an even chance
0.45–0.55	Medium:	> 0.45 <=0.55	An even chance to occur
0.35–0.45	Medium:	> 0.35 <=0.45	Somewhat less than an even chance
0.25–0.35	Low:	> 0.25 <=0.35	Not very likely to occur
0.15–0.25	Low:	> 0.15 <=0.25	Not likely to occur
0.00–0.15	Low:	> 0.00 <=0.15	Almost sure not to occur

ranking (risks with multiple impacts), and risk urgency assessment. Quantitative techniques include weighting of cardinal risk assessments of consequence, probability, and timeframe; probability distributions; sensitivity analysis; expected monetary value analysis; and modeling and simulation. MITRE has developed the min- and max-average approaches (using a weighting scale more heavily weighting the max or min value). Expert judgment is involved in all of these techniques to identify potential impacts, define inputs, and interpret the data [3].

In addition, MITRE has developed the RiskNav® tool that assists managers in assessing and prioritizing program risks. RiskNav includes the ability to weight timeframe in the risk ranking (e.g., how much time to react/potentially mitigate). RiskNav is used by a number of MITRE sponsors and customers. For details on RiskNav, see the article "Risk Management Tools."

Best Practices and Lessons Learned

Tailor the assessment criteria to the decision or project. When assessing risks, recommend techniques and tools that are suitable for the analysis. For example, if the project is an enterprise management or organizational oversight project, then risk impact might be most suitably assessed against goals in lieu of technical performance, cost, and schedule. If the assessment is to determine the risk of investment options, the risk area scale approach might be best suited. For an example of application of risk management, see the Cryptologic Systems Group's Risk Management Implementation Guide [4].

Document the rationale for the assessment of impact and probability. It is important

Table 2. Sample Investment Risk Assessment Scale

	Technical Maturity	Technical Performance	Integration/Interoperability
Details	Maturity of technologies associated with the alternative	Confidence in performance expectations	This refers to Integration and Interoperability (I&I) issues as they affect the alternative's ability to achieve its stated outcome. The extent that I&I is understood has been demonstrated. It is assumed I&I considerations associated.
Low	Key technologies are ready and mature and require little/no effort in time to execute the alternative	There are no technical or performance expectations identified that will have any impact on achieving the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are well understood. Most of the challenging concerns have been resolved and/or successfully tested/demonstrated under representative or actual field conditions. As such, I&I considerations are not expected to have severe negative impact on the ability of this alternative to achieve its stated objectives.
Low med .	Key technologies are expected to be ready and mature in time to execute the alternative	Limited technical or performance expectations identified that will have a minor impact on achieving the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are very well understood. Some challenging concerns have not been resolved and/or successfully tested/demonstrated under representative or actual field conditions. As such, I&I considerations are expected to have negligible impact on the ability of this alternative to achieve its stated objectives.
Med.	Key technologies are not ready and mature and require moderate effort to implement the alternative	Technical or performance limitations have been identified that will have a moderate impact on achieving the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are somewhat-borderline understood. Nearly all (including the most challenging concerns) have been resolved and/or successfully tested/demonstrated under representative or actual field conditions. As such, I&I considerations are expected to have modest negative effects on the ability of this alternative to achieve its stated objectives.
Med.-High	Key technologies are not ready and mature and require significant effort to implement the alternative	There are no technical or performance expectations identified that will have any impact on achieving the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are somewhat-borderline understood. Nearly all (including the most challenging concerns) have been resolved and/or successfully tested/demonstrated under representative or actual field conditions. As such, I&I considerations are expected to have significant negative effects on the ability of this alternative to achieve its stated objectives.

High	Key technologies will not be ready and mature and will have a severe impact on the alternative	Major technical or performance issues have been identified that will have a severe impact on achieving the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are not very well understood. Many challenging concerns are not resolved and/or successfully tested/demonstrated under representative or actual field conditions. As such, I&I considerations are expected to have severe negative effects on the ability of this alternative to achieve its stated objectives.
Catastrophic	Key technologies will not be available and there is no alternative	Serious technical or performance issues have been identified that will prevent achieving any of the stated outcome objectives expected from the alternative	For this alternative, I&I considerations are show-stoppers with the respect to the ability of this alternative to achieve its stated objectives.

to document the justification or rationale for each risk impact assessment and probability of occurrence rating. If the conditions or environment change, the assessment might need to be revisited. The rationale helps to communicate the significance of the risk. When using the investment assessment scale approach, the statement of risk is typically captured in the rationale.

Recognize the role of systems engineering. Risk assessment and management are roles of systems engineering, especially as projects and programs become more complex and interdependent. The judgments that are involved require a breadth of knowledge of system characteristics and the constituent technologies beyond that of design specialists. In addition, the judgments of risk criticality are at the system and program levels [5]. Risk cuts across the life cycle of systems engineering, and MITRE SEs should be prepared to address risk throughout—concept and requirements satisfaction, architectural level risks, design and development risks, training risks, fielding, and environment risks. MITRE SEs are encouraged to advocate for SE involvement in risk assessment and management.

Tailor the prioritization approach to the decision or project. Match the prioritizing algorithm, techniques, and tools to the assessment need (e.g., needs could include time criticality as a prioritization factor, the ability to see capability at risk, the need for a single risk score for the portfolio, the ability to have insight into risks with multiple impacts). Each risk area—threat, operations, programmatic, etc.—will have different priorities. Typically, there will be a priority to these areas themselves—a major threat risk could be totally unacceptable and the effort may be abandoned. If the threat risks are acceptable but the operations cannot be effectively performed, then, again, the effort may be abandoned. Be sure to consider these various decisions and criticality to help the government assess the priorities of mitigating the risks that arise.

Consider MITRE's RiskNav® tool. RiskNav might be appropriate for assessing and prioritizing risks on your government program. MITRE SEs have found that having easy access to this well-tested tool and a support team sometimes encourages government program teams to adopt a more robust risk management process than they otherwise might have. See the article "Risk Management Tools."

Consider Monte Carlo simulations. Monte Carlo simulations use probability distributions to assess the likelihood of achieving particular outcomes, such as cost or completion date [3]. They have been used effectively on a number of MITRE government programs to help the project teams assess schedule risk.

References and Resources

1. The MITRE Institute, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model, Ver. 1," pp. 11, 41–42.
2. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.
3. *A Guide to the Project Management Body of Knowledge*, (PMBOK Guide), 4th Ed., ANSI/PMI 99-001-2008, pp. 273–312.
4. Cryptologic Systems Group, July 2007, *CPSG Risk Management Implementation Guide*, V. 1.2.
5. Kossiakoff, A., and W. N. Sweet, 2003, *Systems Engineering Principles and Practice,* John Wiley and Sons, Inc., pp. 98–106.

Additional References and Resources

Garvey, P. R., January 2000, *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*, Chapman-Hall/CRC Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 0824789660.

International Council on Systems Engineering (INCOSE), January 2010, *INCOSE Systems Engineering Handbook*, Ver.3.2, INCOSE-TP-2003-002-03.2, pp. 213–225.

Lavine, M., S. McBrien, J. Ruddy, and M. Yaphe, August 2006, Methodology for Assessing Alternative IT Architectures for Portfolio Decisions, MITRE Technical Report 06W0000057.

McMahon, C. and R. Henry, September 2009, "RiskAssessment Scales—Sept09." MITRE SEPO, Risk Management Toolkit.

Stevens, R., "Engineering and Acquiring Mega-Systems: Systems Engineering Challenge for the XXI Century," (aka. Megasystems Engineering), June 13, 2007, Presentation to ATEC Professional Development Day.

RISK MANAGEMENT

Risk Mitigation Planning, Implementation, and Progress Monitoring

Definition: *Risk mitigation planning is the process of developing options and actions to enhance opportunities and reduce threats to project objectives [1]. Risk mitigation implementation is the process of executing risk mitigation actions. Risk mitigation progress monitoring includes tracking identified risks, identifying new risks, and evaluating risk process effectiveness throughout the project [1].*

Keywords: *risk, risk management, risk mitigation, risk mitigation implementation, risk mitigation planning, risk mitigation progress monitoring*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) working on government programs develop actionable risk mitigation strategies and monitoring metrics, monitor implementation of risk mitigation plans to ensure successful project and program completion, collaborate with the government team in conducting risk reviews across projects and programs, and analyze metrics to determine ongoing risk status and identify serious risks to elevate to the sponsor or customer [2].

Background

Risk mitigation planning, implementation, and progress monitoring are depicted in Figure 2 in the Risk Management topic article. As part of an iterative process, the risk tracking tool is used to record the results of risk prioritization analysis (step 3) that provides input to both risk mitigation (step 4) and risk impact assessment (step 2).

The risk mitigation step involves development of mitigation plans designed to manage, eliminate, or reduce risk to an acceptable level. Once a plan is implemented, it is continually monitored to assess its efficacy with the intent of revising the course of action if needed.

Risk Mitigation Strategies

General guidelines for applying risk mitigation handling options are shown in Figure 2. These options are based on the assessed combination of the probability of occurrence and severity of the consequence for an identified risk. These guidelines are appropriate for many, but not all, projects and programs.

Risk mitigation handling options include:

- **Assume/Accept:** Acknowledge the existence of a particular risk, and make a deliberate decision to accept it without engaging in special efforts to control it. Approval of project or program leaders is required.
- **Avoid:** Adjust program requirements or constraints to eliminate or reduce the risk. This adjustment could be accommodated by a change in funding, schedule, or technical requirements.
- **Control:** Implement actions to minimize the impact or likelihood of the risk.
- **Transfer:** Reassign organizational accountability, responsibility, and authority to another stakeholder willing to accept the risk.
- **Watch/Monitor:** Monitor the environment for changes that affect the nature and/or the impact of the risk.

Each of these options requires developing a plan that is implemented and

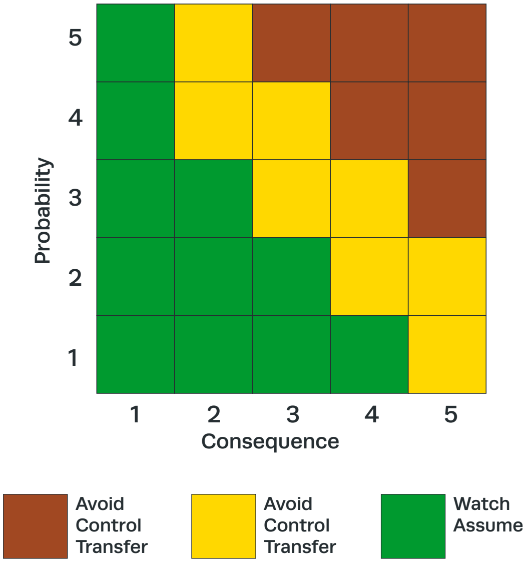


Figure 1. Risk Mitigation Handling Options [3]

monitored for effectiveness. More information on handling options is discussed under Best Practices and Lessons Learned below.

From a systems engineering perspective, common methods of risk reduction or mitigation with identified program risks include the following, listed in order of increasing seriousness of the risk [4]:

1. Intensified technical and management reviews of the engineering process
2. Special oversight of designated component engineering
3. Special analysis and testing of critical design items
4. Rapid prototyping and test feedback
5. Consideration of relieving critical design requirements
6. Initiation of fallback parallel developments

When determining the method for risk mitigation, the MITRE SE can help the customer assess the performance, schedule, and cost impacts of one mitigation strategy over another. For something like "parallel" development mitigation, MITRE SEs could help the government determine whether the cost could more than double, while time might not be extended by much (e.g., double the cost for parallel effort, but also added cost for additional program office and user engagement). For conducting rapid prototyping or changing operational requirements, MITRE SEs can use knowledge in creating prototypes and using prototyping and experimenting (see the article "Special Considerations for Conditions of Uncertainty: Prototyping and Experimentation" and the Requirements Engineering topic) for projecting the cost and time to conduct a prototype to help mitigate particular risks (e.g., requirements). Implementing more engineering reviews and special oversight and testing may require changes to contractual agreements. MITRE systems engineers can help the government assess these (schedule and cost) by helping determine the basis of estimates for additional contractor efforts and providing a reality check for these estimates. MITRE's CASA (Center for Acquisition and Systems Analysis) and the CCG (Center for Connected Government) Investment Management practice department have experience and a knowledge base in many development activities across a wide spectrum of methods and can help with realistic assessments of mitigation alternatives.

For related information, see the other articles in the SEG's Risk Management topic area.

Best Practices and Lessons Learned

Handling Options

Assume/Accept. Collaborate with the operational users to create a collective understanding of risks and their implications. Risks can be character–ized as impacting traditional cost, schedule, and performance parameters. Risks should also be characterized as impact to mission performance resulting from reduced technical performance or capability. Develop an understanding of all these impacts. Bringing users into the mission impact characterization is particularly important to selecting which "assume/accept" option is ultimately chosen. Users will decide whether accepting the consequences of a risk is acceptable. Provide the users with the vulnerabilities affecting a risk, countermeasures that can be performed, and residual risk that may occur. Help the users understand the costs in terms of time and money.

Avoid. Again, work with users to achieve a collective understanding of the implications of risks. Provide users with projections of schedule adjustments needed to reduce risk associated with technology maturity or additional development to improve performance. Identify capabilities that will be delayed and any impacts resulting from dependencies on other efforts. This information better enables users to interpret the operational implications of an "avoid" option.

Control. Help control risks by performing analyses of various mitigation options. For example, one option is to use a commercially available capability instead of a contractor-developed one. In developing options for controlling risk in your program, seek out potential solutions from similar risk situations of other MITRE customers, industry, and academia. When considering a solution from another organization, take special care in assessing any architectural changes needed and their implications.

Transfer. Reassigning accountability, responsibility, or authority for a risk area to another organization can be a double-edged sword. It may make sense when the risk involves a narrow specialized area of expertise not normally found in program offices. But, transferring a risk to another organization can result in dependencies and loss of control that may have their own complications. Position yourself and your customer to consider a transfer option by acquiring and maintaining awareness of organizations within your customer space that focus on specialized needs and their solutions. Acquire this awareness as early in the program acquisition cycle as possible, when transfer options are more easily implemented.

Watch/monitor. Once a risk has been identified and a plan put in place to manage it, there can be a tendency to adopt a "heads down" attitude, particularly if the execution of the mitigation appears to be operating on "cruise control." Resist that inclination. Periodically revisit the basic assumptions and premises of the risk. Scan the environment to see whether the situation has changed in a way that affects the nature or impact of the risk. The risk may have changed sufficiently so that the current mitigation is ineffective and needs to be scrapped in favor of a different one. On the other hand, the risk may have diminished in a way that allows resources devoted to it to be redirected.

Determining Mitigation Plans

Understand the users and their needs. The users/operational decision makers will be the decision authority for accepting and avoiding risks. Maintain a close relationship with the user community throughout the systems engineering life cycle. Realize that mission accomplishment is paramount to the user community and acceptance of residual risk should be firmly rooted in a mission decision.

Seek out the experts and use them. Seek out the experts within and outside MITRE. MITRE's technical centers exist to provide support in their specialty areas. They understand what's feasible, what's worked and been implemented, what's easy, and what's hard. They have the knowledge and experience essential to risk assessment in their area of expertise. Know our internal centers of excellence, cultivate relationships with them, and know when and how to use them.

Recognize risks that recur. Identify and maintain awareness of the risks that are "always there"—interfaces, dependencies, changes in needs, environment and requirements, information security, and gaps or holes in contractor and program office skill sets. Help create an acceptance by the government that these risks will occur and recur and that plans for mitigation are needed up front. Recommend various mitigation approaches, including adoption of an evolution strategy, prototyping, experimentation, engagement with broader stakeholder community, and the like.

Encourage risk taking. Given all that has been said in this article and its companions, this may appear to be an odd piece of advice. The point is that there are consequences of not taking risks, some of which may be negative. Help the customer and users understand that reality and the potential consequences of being overly timid and not taking certain risks in your program. An example of a negative consequence for not taking a risk when delivering a full capability is that an adversary might realize a gain against our operational users. Risks are not defeats, but simply bumps in the road that need to be anticipated and dealt with.

Recognize opportunities. Help the government understand and see opportunities that may arise from a risk. When considering alternatives for managing a particular risk, be sure to assess whether they provide an opportunistic advantage by improving performance, capacity, flexibility, or desirable attributes in other areas not directly associated with the risk.

Encourage deliberate consideration of mitigation options. This piece of advice is good anytime, but particularly when supporting a fast-paced, quick reaction government program that is juggling many competing priorities. Carefully analyze mitigation options and encourage thorough discussion by the program team. This is the form of the wisdom "go slow to go fast."

Not all risks require mitigation plans. Risk events assessed as medium or high criticality should go into risk mitigation planning and implementation. On the other hand, consider whether some low criticality risks might just be tracked and monitored on a watch list. Husband your risk-related resources.

Mitigation Plan Content

Determine the appropriate risk manager. The risk manager is responsible for identifying and implementing the risk mitigation plan. He or she must have the knowledge, authority, and resources to implement the plan. Risk mitigation activities will not be effective without an engaged risk manager. It may be necessary to engage higher levels in the customer organization to ensure the need for the risk manager is addressed. This can be difficult and usually involves engaging more senior levels of the MITRE team as well.

Develop a high-level mitigation strategy. This is an overall approach to reduce the risk impact severity and/or probability of occurrence. It could affect a number of risks and include, for example, increasing staffing or reducing scope.

Identify actions and steps needed to implement the mitigation strategy. Ask these key questions:

What actions are needed?

- Make sure you have the right exit criteria for each. For example, appropriate decisions, agreements, and actions resulting from a meeting would be required for exit, not merely the fact that the meeting was held.
- Look for evaluation, proof, and validation of met criteria. Consider, for example, metrics or test events.
- Include only and all stakeholders relevant to the step, action, or decisions.

When must actions be completed?

- Backward Planning: Evaluate the risk impact and schedule of need for the successful completion of the program and evaluate test events, design considerations, and more.
- Forward Planning: Determine the time needed to complete each action step and when the expected completion date should be.
- Evaluate key decision points and determine when a move to a contingency plan should be taken.

Who is the responsible action owner?

- What resources are required? Consider, for example, additional funding or collaboration.
- How will this action reduce the probability or severity of impact?

Develop a contingency plan ("fall back, plan B") for any high risk.

- Are cues and triggers identified to activate contingency plans and risk reviews?
- Include decision point dates to move to fallback plans. The date to move must allow time to execute the contingency plan.

Evaluate the status of each action. Determine when each action is expected to be completed successfully.

Integrate plans into IMS and program management baselines. Risk plans are integral to the program, not something apart from it.

Monitoring Risk

Include risk monitoring as part of the program review and manage continuously. Monitoring risks should be a standard part of program reviews. At the same time, risks should be managed continuously rather than just before a program review. Routinely review plans in management meetings.

Review and track risk mitigation actions for progress. Determine when each action is expected to be completed successfully.

Refine and redefine strategies and action steps as needed.

Revisit risk analysis as plans and actions are successfully completed. Are the risks burning down? Evaluate impact to program critical path.

Routinely reassess the program's risk exposure. Evaluate the current environment for new risks or modification to existing risks.

References and Resources

1. Project Management Institute, *A Guide to the Project Management Body of Knowledge, (PMBOK Guide),* Fourth Edition, ANSI/PMI 99-001-2008, pp. 273–312.
2. The MITRE Institute, September 1, 2007, MITRE Systems Engineering (SE) Competency Model, Ver. 1, pp. 10, 40–41.
3. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.
4. Kossiakoff, A. and W. N. Sweet, 2003*, Systems Engineering Principles and Practice,* John Wiley and Sons, Inc., pp. 98–106.

Additional References and Resources

International Council on Systems Engineering (INCOSE), January 2010, *INCOSE Systems Engineering Handbook*, Ver. 3.2, INCOSE-TP-2003-002-03.2, pp. 213–225.

Definition: *Risk management tools support the implementation and execution of program risk management in systems engineering programs.*

Keywords: *risk analysis tools, risk management tools, risk tools*

RISK MANAGEMENT

Risk Management Tools

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) working on government programs are expected to use risk analysis and management tools to support risk management efforts. MITRE SEs also are expected to understand the purpose, outputs, strengths, and limitations of the risk tool being used.

Background

Risk analysis and management tools serve multiple purposes and come in many shapes and sizes. Some risk analysis and management tools include those used for:

- **Strategic and Capability Risk Analysis:** Focuses on identifying, analyzing, and prioritizing risks to achieve strategic goals, objectives, and capabilities.
- **Threat Analysis:** Focuses on identifying, analyzing, and prioritizing threats to minimize their impact on national security.
- **Investment and Portfolio Risk Analysis:** Focuses on identifying, analyzing, and prioritizing investments and possible alternatives based on risk.
- **Program Risk Management:** Focuses on identifying, analyzing, prioritizing, and managing risks to eliminate or minimize their impact on a program's objectives and probability of success.
- **Cost Risk Analysis:** Focuses on quantifying how technological and economic risks may affect a system's cost. Applies probability methods to model, measure, and manage risk in the cost of engineering advanced systems.

Each specialized risk analysis and management area has developed tools to support its objectives with various levels of maturity. This article focuses on tools that support the implementation and execution of program risk management.

Selecting the Right Tool

It is important that the organization defines the risk analysis and management process before selecting a tool. Ultimately, the tool must support the process. Consider the following criteria when selecting a risk analysis and management tool:

- **Aligned to risk analysis objectives:** Does the tool support the analysis that the organization is trying to accomplish? Is the organization attempting to implement an ongoing risk management process or conduct a one-time risk analysis?
- **Supports decision making:** Does the tool provide the necessary information to support decision making?
- **Accessibility:** Is the tool accessible to all users and key stakeholders? Can the tool be located/hosted where all necessary personnel can access it?
- **Availability of data:** Is data available for the tool's analysis?
- **Level of detail:** Is the tool detailed enough to support decision making?
- **Integration with other program management/systems engineering processes:** Does the tool support integration with other program management/systems engineering processes?

Program Risk Management Tools

In program risk management, it is important to select a tool that supports the risk management process steps outlined in Figure 1 in the preceding Risk Management topic article. See the other articles in the SEG's Risk Management topic area for additional information on each of the process steps. Many tools are available that support the implementation of program risk management. Many tools also can be used to support the management of project, enterprise, and system-of-systems risks.

MITRE Developed Tools

RiskNav®

RiskNav® (RiskNav is a registered trademark of The MITRE Corporation) is a well-tested tool developed by MITRE to facilitate the risk process and help program managers handle their risk space. RiskNav lets you collect, analyze, prioritize, monitor, and visualize risk information in a collaborative fashion. This tool provides three dimensions of information graphically—risk priority, probability, and mitigation/management status.

RiskNav, originally produced for the U.S. government, is designed to capture, analyze, and display risks at a project or enterprise level. RiskNav is currently deployed throughout numerous MITRE sponsors or clients.

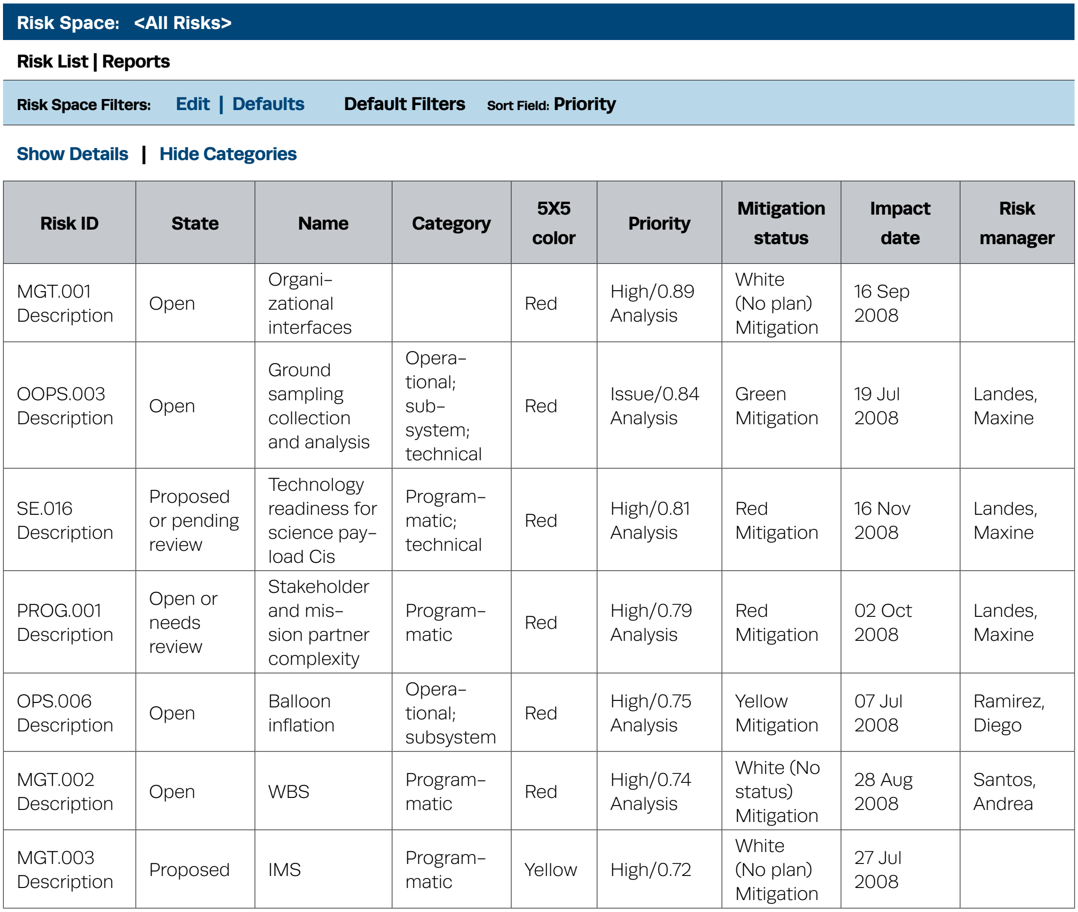
Since January 2005, the Technology Transfer Office at MITRE has licensed RiskNav technology to commercial companies. Current licensees include Sycamore.US, Inc. and NMR Consulting. The Technology Transfer Office will support the tool for contractor and other government acquisition, and will ensure that proper licensing forms are obtained and signed by new users. There is no cost for government usage. This formal procedure is not needed if MITRE is hosting a risk management effort.

RiskNav presents the risk space in tabular and graphical form. The tabular form, shown in Table 1, presents key information for each risk, and allows the risk space to be filtered and sorted to focus on the most important risks. The information in the tables and figures is artificial and for illustrative purposes only. It does not represent real programs, past or present.

RiskNav uses a weighted average model (Table 2) that computes an overall score for each identified risk. The risk priority is a weighted average of the timeframe (how soon the risk will occur), probability of occurrence, and impact (cost, schedule, technical). This score provides a rank order of the risks from most critical to least critical. Formally, this scoring model originates from the concept of linear utility, where more important risks get higher numbers, and the gaps between the numbers correspond to the relative strengths of the differences.

In graphical form (Figure 1), RiskNav represents three key aspects of each risk in the risk space—risk priority, probability, and the mitigation/management status. The data points

Table 1. RiskNav Summaries of Key Risk Information



Risk Space: <All Risks>

Risk List | Reports

Risk Space Filters: **Edit | Defaults** **Default Filters** **Sort Field: Priority**

Show Details | Hide Categories

Risk ID	State	Name	Category	5X5 color	Priority	Mitigation status	Impact date	Risk manager
MGT.001 Description	Open	Organizational interfaces		Red	High/0.89 Analysis	White (No plan) Mitigation	16 Sep 2008	
OOPS.003 Description	Open	Ground sampling collection and analysis	Operational; subsystem; technical	Red	Issue/0.84 Analysis	Green Mitigation	19 Jul 2008	Landes, Maxine
SE.016 Description	Proposed or pending review	Technology readiness for science payload Cis	Programmatic; technical	Red	High/0.81 Analysis	Red Mitigation	16 Nov 2008	Landes, Maxine
PROG.001 Description	Open or needs review	Stakeholder and mission partner complexity	Programmatic	Red	High/0.79 Analysis	Red Mitigation	02 Oct 2008	Landes, Maxine
OPS.006 Description	Open	Balloon inflation	Operational; subsystem	Red	High/0.75 Analysis	Yellow Mitigation	07 Jul 2008	Ramirez, Diego
MGT.002 Description	Open	WBS	Programmatic	Red	High/0.74 Analysis	White (No status) Mitigation	28 Aug 2008	Santos, Andrea
MGT.003 Description	Proposed	IMS	Programmatic	Yellow	High/0.72	White (No plan) Mitigation	27 Jul 2008	

represent risks, and the color of a box indicates the status of the mitigation action (*White*: no plan; *Red*: plan not working; *Yellow*: may not work; *Green*: most likely successful; *Blue*: completed successfully; *Black*: actions will start at a later date). Data points can be selected to show detailed risk information about the analysis, who is working the management actions, the status, and other information.

RiskNav also displays a 5x5 frequency chart (Figure 2) showing the number of risks in each square of a 5x5 matrix of probability versus consequence ranges. The *Red* cells contain the highest priority risks. The *Yellow* and *Green* cells contain medium and low priority risks, respectively. RiskNav incorporates an administrative capability that allows the chart's probability and consequence ranges to be customized. Clicking on a cell provides a detailed list of

Table 2. RiskNav Uses a Scoring Model to Prioritize Risks

Risk Analysis Inputs		Computed Risk Scores	
Impact Date:	16 Sep 2008	Risk Timeframe:	Short-term / 0.99
Probability:	High / 0.90	Overall Risk Impact:	High / 0.79
Cost Impact Rating:	High / 0.83	Risk Consequence:	High / 0.89
Schedule Impact Rating:	High / 0.83	Risk Priority:	High / 0.89
Technical Impact Rating:	High / 0.65	Risk Ranking: (Ranks "open" risks with priority > 0)	
Compliance & Oversight Impact Rating:	High / 0.83	Rank in Program:	1 of 17
		Rank in Organization:	1 of 4
		Rank in Project:	1 of 2

the risks in that cell. The All Red, All Yellow, *and* All Green icons at the top of the chart can be used to list risks in all cells of a particular color.

RiskNav is a Web application that runs on a personal computer (PC) server, which can concurrently be used as a client. Once installed, it is intended to run using Internet Explorer as the browser.

Because RiskNav is a Web application, its installation requires more experience than simply installing a normal executable. A detailed installation guide is available to assist in the installation process. However, it is assumed that the installer has expertise installing and configuring Windows Web-based software. To obtain more information about RiskNav, email risknav@mitre.org.

Risk Matrix

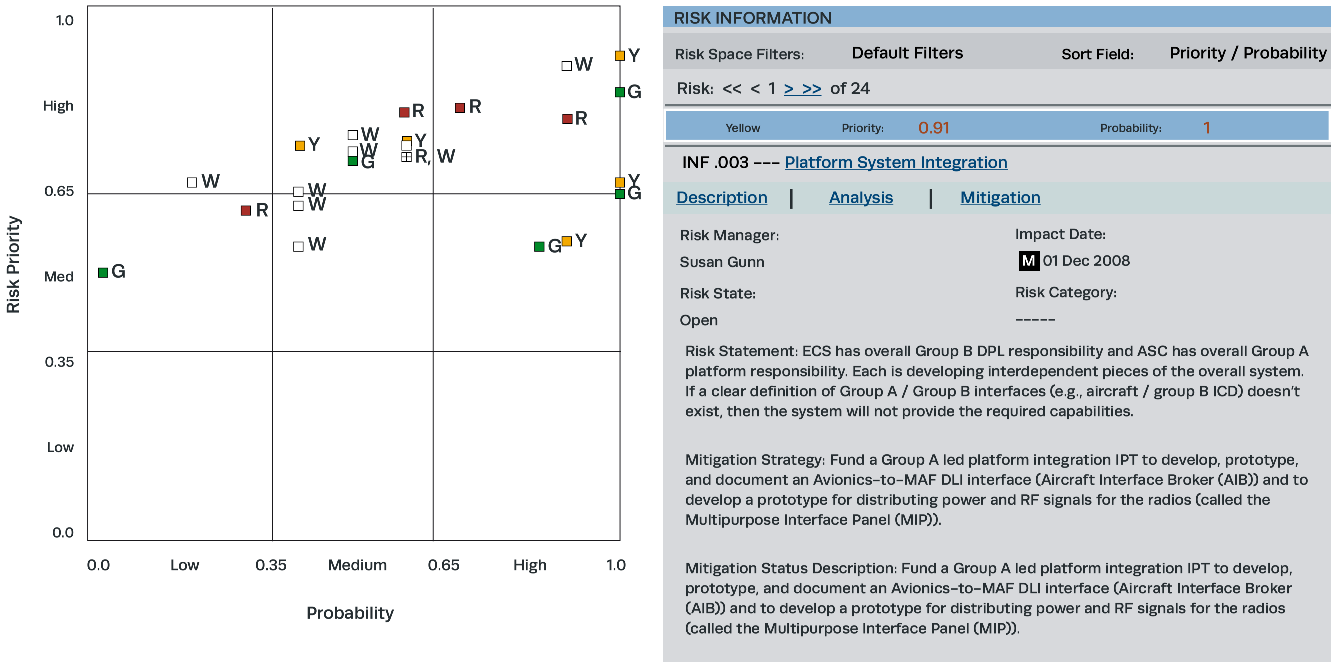
Risk Matrix is a software application that can help identify, prioritize, and manage key risks on a program. MITRE created it a few years ago to support a risk assessment process developed by a MITRE DoD client. MITRE and the client have expanded and improved the original process, creating the Baseline Risk Assessment Process. Although the process and application were developed for use by a specific client, these principles can be applied to most government acquisition projects. (See Figure 3.)

Risk Matrix (as well as more information on RiskNav and Risk Radar) is available in the Systems Engineering Process Office (SEPO) Risk Management Toolkit. Although Risk Matrix is available for public release, support is limited to downloadable online documentation.

Commercial Tools

Many commercial tools are available to support program risk management efforts. Risk management tools most commonly used by the government are:

- Risk Radar and Risk Radar Enterprise— American Systems
- Active Risk Manager— Strategic Thought Group



RISK INFORMATION

Risk Space Filters: Default Filters Sort Field: Priority / Probability

Risk: << < 1 > >> of 24

Yellow Priority: 0.91 Probability: 1

INF .003 --- Platform System Integration

Description | Analysis | Mitigation

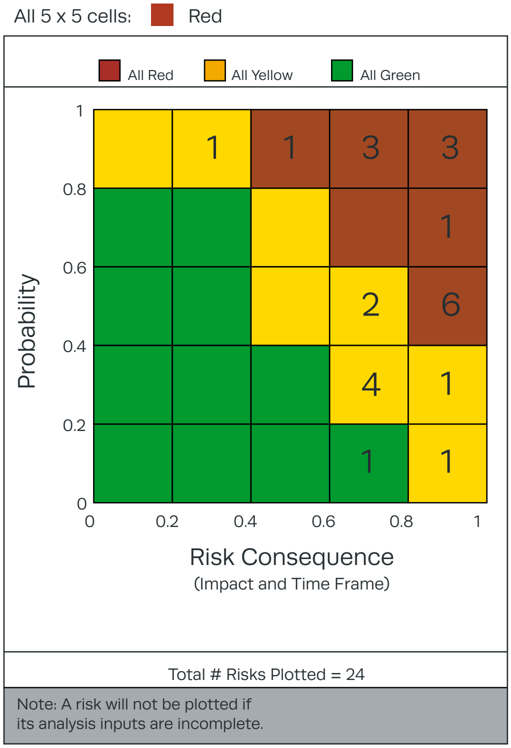
Risk Manager:	Impact Date:
Susan Gunn	M 01 Dec 2008
Risk State:	Risk Category:
Open	-----

Risk Statement: ECS has overall Group B DPL responsibility and ASC has overall Group A platform responsibility. Each is developing interdependent pieces of the overall system. If a clear definition of Group A / Group B interfaces (e.g., aircraft / group B ICD) doesn't exist, then the system will not provide the required capabilities.

Mitigation Strategy: Fund a Group A led platform integration IPT to develop, prototype, and document an Avionics-to-MAF DLI interface (Aircraft Interface Broker (AIB)) and to develop a prototype for distributing power and RF signals for the radios (called the Multipurpose Interface Panel (MIP)).

Mitigation Status Description: Fund a Group A led platform integration IPT to develop, prototype, and document an Avionics-to-MAF DLI interface (Aircraft Interface Broker (AIB)) and to develop a prototype for distributing power and RF signals for the radios (called the Multipurpose Interface Panel (MIP)).

Figure 1. RiskNav Visualizes the Risk Space Showing Risk Priority and Mitigation Status



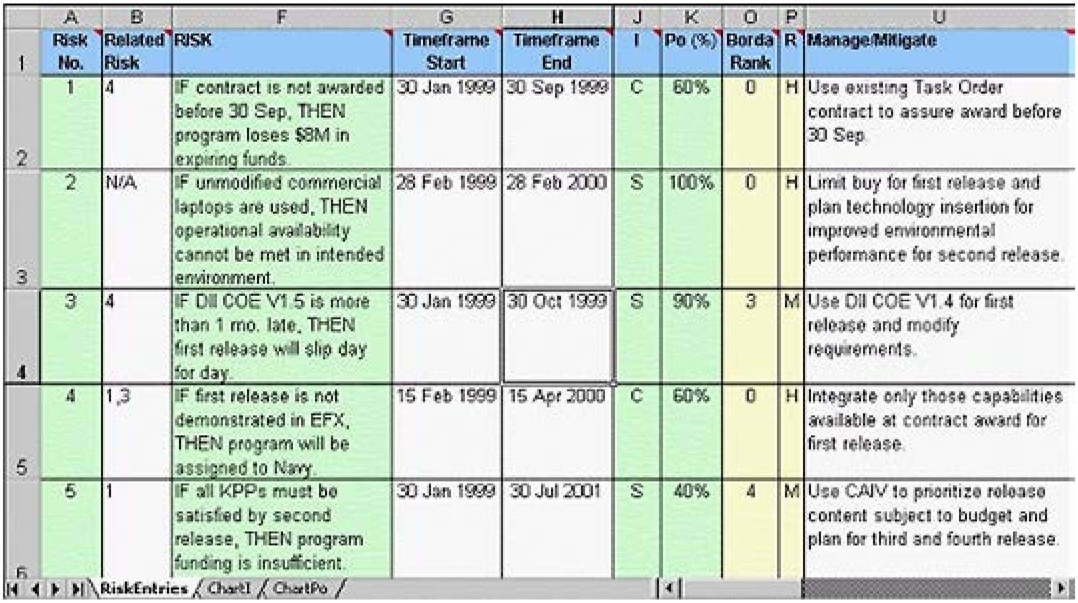
Risk ID	Risk Name	Consequence	Probability	Mitigation
INF.003	Platform system integration	0.91	100	Yellow
MGT.001	Organizational interfaces	0.89	0.90	White
OPS.003	Ground sampling collection and analysis	0.84	100	Green
SE.016	Technology readiness for science payload CIs	0.87	0.70	Red
SE.015	Cannot handle real–time transmission rate	0.91	0.60	Red
PROG.001	Stakeholder and mission partner complexity	0.73	0.90	Red
SE.017	Balloon weight	0.89	0.50	White
OPS.006	Balloon inflation	0.82	0.60	Yellow
MGT.002	WBS	0.81	0.60	White
OPS.009	Accurately controlling vehicle altitude in the Venusian environment	0.84	0.50	Yellow
SE.005	Component single point of failure	0.81	0.50	Green
APPS.001	SW testing environment	0.67	100	Yellow

Figure 2. 5x5 Frequency Chart to Identify High–Priority Risks

Both tools are Web-based applications that support all steps in the risk management process.

Contractor Tools

Government programs sometimes implement a combined government/contractor risk management process that uses tools provided by the contractor. Multiple major government contractors have developed in-house risk management applications. Many applications are comparable to MITRE and commercial tools available, and effectively support program risk management.



	A	B	F	G	H	J	K	O	P	U
1	Risk No.	Related Risk	RISK	Timeframe Start	Timeframe End	I	Po (%)	Borda Rank	R	Manage/Mitigate
2	1	4	IF contract is not awarded before 30 Sep, THEN program loses $8M in expiring funds.	30 Jan 1999	30 Sep 1999	C	60%	0	H	Use existing Task Order contract to assure award before 30 Sep.
3	2	N/A	IF unmodified commercial laptops are used, THEN operational availability cannot be met in intended environment.	28 Feb 1999	28 Feb 2000	S	100%	0	H	Limit buy for first release and plan technology insertion for improved environmental performance for second release.
4	3	4	IF DII COE V1.5 is more than 1 mo. late, THEN first release will slip day for day.	30 Jan 1999	30 Oct 1999	S	90%	3	M	Use DII COE V1.4 for first release and modify requirements.
5	4	1,3	IF first release is not demonstrated in EFX, THEN program will be assigned to Navy.	15 Feb 1999	15 Apr 2000	C	60%	0	H	Integrate only those capabilities available at contract award for first release.
6	5	1	IF all KPPs must be satisfied by second release, THEN program funding is insufficient.	30 Jan 1999	30 Jul 2001	S	40%	4	M	Use CAIV to prioritize release content subject to budget and plan for third and fourth release.

Figure 3. Screenshot of Risk Matrix

Customized Tools

Many smaller programs use Microsoft Excel or Access customized risk management tools. Some customized solutions meet the tool selection criteria outlined above. This is important when considering a customized solution that meets the need of the program being supported.

Best Practices and Lessons Learned

Fit the tool to the process or assessment needed. There are many types of risk analysis and management tools available, including ones for financial analysis, cost-risk uncertainty, and traditional program management. Understand the need of the program, reporting, analysis (e.g., ability to modify risk impact scales to reflect the need), and accessibility (e.g., multiple user environment), before selecting a tool. Do not let the tool drive the process.

Change the tool if it does not support decision making and the process. As the risk process matures and reporting needs evolve, it is important to change the risk management tool used to support this changed environment. The following conditions could warrant a change in the risk management tool:

- **New reporting requirements:** It is best to use a tool that matches reporting requirements.

- **Increase in level of mitigation detail needed:** Some tools capture only high-level mitigation plans, whereas others capture detailed plans with action steps and statuses.
- **Team capacity unable to support tool:** If the tool is too burdensome, it is important to examine ways to streamline its use or change to another tool that better supports the program's environment.

Maximize access to the tool. It is important that the widest cross-section of the team has access to the tool and is responsible for updates. This ensures distribution of workload and ownership, and prevents bottlenecks in the process.

References and Resources

1. Garvey, P. R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective,* Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), ISBN: 1584886374.

Additional References and Resources

Garvey, P. R., January 2000, *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective,* Chapman-Hall/CRC Press, Taylor & Francis Group (UK), ISBN: 0824789660.

MITRE E520 Risk Analysis and Management Technical Team, "Risk Analysis and Management Tools."

Risk Process Guidance, SEPO Risk Management Toolkit.

Configuration Management

Definition: *Configuration management is the application of sound program practices to establish and maintain consistency of a product's or system's attributes with its requirements and evolving technical baseline over its life. It involves interaction among government and contractor program functions in an integrated product team environment. A configuration management process guides the system products, processes, and related documentation, and facilitates the development of open systems. Configuration management efforts result in a complete audit trail of plans, decisions, and design modifications [1].*

Keywords: *acquisition development program, program control configuration management policy, program management*

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) should have a sound understanding of the principles of configuration management (CM), how the development organization initiates configuration control, how the developer implements configuration management, and how the government sponsor or sustainment organization continues the configuration management following product delivery. Because many of our programs place a significant amount of their configuration management effort on software configuration management and commercial hardware and software, that will be the focus of this discussion; however, the configuration management of developmental hardware deserves a similar discussion. Usually MITRE's role in the practice of configuration management is to ensure that good CM processes are in place, documented as part of a program Systems Engineering Plan and/or Life-cycle Management Plan, and followed by all contractors and program office personnel. The implementation of the process is not likely to be a MITRE role, unless there are special circumstances (e.g., analysis of a CM breakdown). Issues such as the use of appropriate CM tools for a development environment, application of automated system configuration monitoring, and the frequent conundrum of managing moving baselines are likely to be the focus of MITRE's expertise for CM.

Context

Why do we perform configuration management? One reason is that it is required on Department of Defense (DoD) [2], Federal Aviation Administration (FAA) [3], Internal Revenue Service (IRS) [4], and other formal organizationally run programs; behind those requirements are the reasons and lessons learned. Once a program has been developed and deployed as a system, it is necessary to understand the baseline for reasons of sustainability and affordability. Obviously it is very hard to maintain or upgrade something that is undefined and not controlled. We need to minimize unintended negative consequences and cost of changes that are not fully analyzed. We would prefer to have "interchangeable parts," and standards and baselines that allow for ease of maintenance and interoperability. Finally, there is the need for repeatable performance for operations, test, security, and safety. Think of the ability to react to an Information Assurance Vulnerability Assessment when you do not know what could be affected in your system baseline or where the items are located. Think of your ability to specify a replacement system when you do not know what constitutes your system or configuration item baselines because those changes were not tracked after the initial system deployment.

Best Practices and Lessons Learned

Consistent with the systems engineering life cycle, configuration management exists for the life of a program and system. As part of initial program planning, MITRE is involved in the establishment of systems engineering processes, one of which is configuration management.

A plan is essential. A configuration management plan is necessary for sound configuration management practice. Include the following in the plan:

- **Configuration identification.** Identify the things to be managed and level of control at each level.
  - Identify all configuration items to be controlled: user requirements documents, requirements specifications and

traceability, design artifacts, development documents, software version documents, interface control documents, drawings and parts lists, test plans and procedures, test scripts, test results, training materials; depending on the type of program, you may also have architecture products, data flows and network diagrams, simulation data, test harness/modeling and simulations, etc.

  - Identify the level of detail of each to be controlled: system-of-systems, system, configuration item, component, item, part number, network asset, etc.
  - Identify all baselines to be managed: user requirements, system requirements, design, development, test, sustainment, experimentation, etc.
  - Develop a schema or comply with organizational policy to provide unique identifiers for each item.
  - Determine the level of the configuration management hierarchy (stakeholders) for each identified "configuration item" to be approved (baselined).

- **Configuration control.**
  - Develop a closed-loop corrective action process to track all configuration item changes to closure and inclusion in appropriate baseline documentation.
  - Build or provide specifications to build work products from the software configuration management system or physical products from the hardware configuration management system.
  - Purchase or develop tools for version control of source code. This product should provide version control tracking to the line of code level. Assure implementation of an engineering release system to provide hardware version control.
- **Configuration status accounting.** Publish periodic reports describing the current configuration of each configuration item. There should be a configuration version description document detailing each version of software undergoing integration, system, or acceptance test. There should be a set of engineering drawings detailing each developmental hardware item undergoing integration and testing. Commercial hardware and software also needs to be under configuration control during integration and testing. Configuration status accounting applies to all fielded hardware, software, and other controlled assets during operations and maintenance for the life of the system.
- **Configuration audits.** Perform periodic examinations of operational baselines for completeness (configuration verification audit). Prior to product delivery to the sponsor, ensure successful completion of a functional configuration audit to assure that the product meets its specified requirements. Also conduct a physical configuration audit to assure that the successfully tested product matches the documentation.
- **Accounting of requirements changes** per month and changes processing time; also, the number of defects that are open and closed are metrics that may be used for configuration management.

Automate to manage complexity. If the program is sufficiently complex, identify and install an automated tool to support the configuration management tasks. Consider the other stakeholders (engineers/programmers, users, contractors, interfacing systems, and sustainment organizations) in the selection of any automated configuration management tools.

Work your plan. Implement and conduct the configuration management activities according to the program's configuration management plan.

Use checklists. A basic checklist, such as the one below, can assist in capturing the necessary efforts.

Table 1. Checklist

Checklist Item	
Have all items subject to configuration control been identified in the program plan?	
Has a closed loop change management system been established and implemented?	
Has a government configuration control board been established for both development and sustainment?	
Are impact reviews performed to ensure that the proposed changes have not comprised the performance, reliability, safety, or security of the system?	
Does the developer's CM create or release all baselines for internal use?	
Does the developer's CM create or release all baselines for delivery to the customer?	
Are records established and maintained describing configuration items?	
Are audits of CM activities performed to confirm that baselines and documents are accurate?	
Do sponsor, program office, primary developer team, and sustainment organizations have CM systems and expertise? Are developers and managers trained equivalently on CM?	
Are CM resources across development team interoperable and compatible (i.e., use of SourceSafe, CVS, CAD/CAM, Requirements Management, and Subversion may represent logistical issues if left unmanaged)?	

MITRE's experience has shown that the CM process chosen for a particular program may be a rigorous, serial process to tightly control the system baseline (i.e., for an aircraft where flight safety is paramount) or a more flexible process that allows for urgent modifications and variable levels of approval. The challenge is to determine the process that best meets stakeholder needs as well as the acquisition/procurement/maintenance needs of the system program office. Depending on the level of complexity, the number of stakeholders, and the nature of the system (e.g., system-of-systems), see the Engineering Information-Intensive Enterprises topic in the SEG's Enterprise Engineering section. Enterprise management, evolutionary acquisition, and spiral development combined with sustainment introduce some unique challenges in configuration management, because of the number of concurrent development and operational baselines that must be controlled and maintained for development, test, experimentation, and operations. Understanding the complexity of the system will enable you to apply the appropriate CM process and the relationship between layers of the CM hierarchy. See the article "How to Control a Moving Baseline."

The number and types of tools employed to assist in configuration management have grown and changed according to technology and development techniques. Once the list of items to be configuration controlled has been determined, assess the variety of tools appropriate to automate the management and control process (i.e., Dynamic Object-Oriented Requirements System tool for requirements management and traceability, Deficiency Reporting Databases, Software Configuration Management tools, Network discovery tools, etc.). For additional guidance, see the article "Configuration Management Tools."

References and Resources

1. Acquisition Community Connection, Defense Acquisition Guidebook, 4.2.3.1.6, accessed January 25, 2010.
2. DoD, December 2008, DoDI 5000.02.
3. FAA, August 7, 2008, FAA Order 1800.66, Configuration Management Policy.
4. Department of Treasury (IRS), July 1, 2007, Enterprise Life Cycle Guide, Process Management and Improvement Policy.

Additional References and Resources

ANSI/EIA-649-A, 2004, National Consensus Standard for Configuration Management.

Association for Configuration and Data Management website, accessed January 25, 2010.

CM Crossroads, "Configuration Management Yellow Pages," accessed January 25, 2010.

IEEE Standards Association, "IEEE Std 828-2005 IEEE Standard for Software Configuration Management Plans," accessed January 25, 2010.

MIL-HDBK-61A, February 2001, Configuration Management Guidance, accessed January 25, 2010.

MITRE Center for Connected Government, CM Toolbox.

MITRE Systems Engineering Practice Office, Configuration Management Toolkit.

Software Technology Support Center, “Configuration Management,” accessed February 2, 2010.

Ventimiglia, B., February 1998, “Effective Software Configuration Management,” *CrossTalk*.

Definition: *Configuration Management (CM) is the application of sound practices to establish and maintain consistency of a product or system's attributes with its requirements and evolving technical baseline over its life [1].*

Keywords: *CM, CM process, configuration baseline, configuration management, modification management, moving baseline, UCN, urgent compelling need*

CONFIGURATION MANAGEMENT

How to Control a Moving Baseline

MITRE SE Roles and Expectations: Although configuration management is not a primary focus of MITRE's systems engineering, it is an integral part of the overall systems engineering job and is an area MITRE systems engineers (SEs) are expected to understand sufficiently. SEs need to monitor and evaluate the CM efforts of others and recommend changes when warranted.

One of the more challenging tasks for the SE of a particular program is to assure the application of sound CM practices in environments where change is pervasive and persistent.

Increasingly the types of programs that MITRE supports are becoming more complex and reliant on commercial technology. Many of our customers' users are facing asymmetric threats and constantly changing mission environments.

The result is that the programs we work on have constantly changing baselines that need to be managed efficiently and effectively. This is most evident in programs that are already fielded and have substantial development efforts taking place in parallel. In these situations, the government is responsible for the overall CM of the program (fielded baseline as well as oversight of development efforts being done by industry).

Best Practices and Lessons Learned

CM is "paperwork"—the "right paperwork." Tracking and managing a fielded configuration baseline is largely an exercise in documentation management. The configuration baseline is an amalgamation of various documents (hardware drawings, software version descriptions documents, interface control/design documents, technical orders, etc.), and is normally managed as site-specific configurations for fielded systems. Development contractors and vendors maintain configuration control of their various software products and applications that make up the system, but the overall configuration baseline is at the system level and managed by the customer organization or program office. From this perspective, configuration management really means maintaining positive control over the paperwork that describes the configuration. The hard part is to determine what level of documentation provides the most accurate representation of the configuration. Experience has shown that the hardware drawings, software version descriptions, technical orders, and systems specifications provide the most bang for the CM buck.

"Shock treatment CM" can be useful—when used sparingly. CM normally works well up until programs are fielded and delivered to operating locations. After fielding and over time, CM can break down. Users at different operating locations have an interest in the system doing specifically what they need, and needs between differing locations can vary enough that the baselines begin to diverge. This can also happen in development programs when different versions of a system are delivered to different users with slightly different baselines. This poses a fundamental dilemma and choice for CM. Is the goal to manage a common core baseline with divergent parts for different users and locations? Or is the goal to maintain a single baseline (for interoperability or standard operations and training reasons, for example)? Either answer can be right. The problem arises when there is no explicit discussion and decision. The usual consequence is that CM breaks down because the different customers and locations start to assume they have control and modify the baseline themselves.

How can this situation be remedied? Sometimes "shock treatment" is the only way to re-assert CM authority. Essentially, this is the threat of closing down an operation due to critical certification or security baselines not being upheld. It is an extreme measure, but if the operation of the system is critical enough to warrant consistent certification of operations or security, it can be a useful hammer. As an example, in one government program, the Designated Accreditation Authority (DAA) became aware of baseline control deviations that resulted in concerns over the integrity of the system. Sites (operating locations) were put on notice by the DAA that their security accreditation would be jeopardized if they did not adhere to the configuration management controls instituted by the program office. This strict enforcement technique sent shock waves throughout the user community, the sustainment organization, and the program office. Uncontrolled baseline modifications soon became a thing of the past. "Shock treatment" CM can be a useful tool to get a configuration back on track. But, it is not an enduring solution to CM issues.

CM—a balance between rigor and reality. CM processes tend to swing back and forth like a pendulum in meticulousness, resources applied, and adherence to documented process. There are benefits from a highly disciplined CM process. However, an unintended consequence can be delays in processing baseline changes and ultimately in fielding modifications. When this happens, the phrase, "CM is slowing me down" may become common. Experience has shown that the most effective CM processes strike a balance between sufficient standards and control processes and mission needs of the program/system. Consider the following factors when determining your CM process: life-cycle stage, operational environment, acceptable risk, and mission requirements. The volume of data maintained is not necessarily a good metric for measuring the effectiveness of a CM process. Less can be better, especially if the quality of data meets your "good enough" threshold.

Get the user invested in the CM process. The user should be your most critical stakeholder and strongest advocate in quality and disciplined CM. Their early and active involvement throughout the CM process is a must. Understandably, users tend to favor speed of execution, and they commonly consider CM as a burden to be managed by engineers. When operational users participate as members on modification teams, engineering reviews, and configuration control boards, they become vested, resulting in a CM process owned by both the managing program office and the user. Users can feel "stonewalled" when they ask for a change that never appears to happen, if the process appears chaotic or inefficient, or is just not understood. An operational user properly invested in CM is the best advocate or spokesperson to the rest of the user community for a disciplined CM process.

Two compelling examples come from an existing government program. In both cases, the user representatives proved to be critical to the development and implementation of two major CM process changes for the system. The first was the establishment of an urgent and compelling need (UCN) process to facilitate rapid fielding of mission critical modifications. Working together, the systems engineers and the operational users negotiated and designed a minimum acceptable CM (documentation and approval) process that met the "good enough" standards for program management, engineering, and the users. This process has stood the test of time, and has proven itself over several years of direct support to field operations. The second example involved the transition to a bulk release modification process for fielding modifications. The bulk release method significantly reduced the number of disruptions and logistical burdens at each site by combining non-time-critical major and minor modifications into one integrated baseline release, with the result that the overall time spent at the sites for modifications was reduced.

Complexity of enterprise CM. Asserting CM control over a small system with clearly defined interfaces can be fairly straightforward. In general, applying good CM practices to that case is well understood. But as programs transition from being primarily "stovepipe" configurations to systems that are part of an enterprise, the scope of their CM processes changes as well. Enterprise systems have more stakeholders to involve in the process, and they also tend to have more moving parts, making CM particularly challenging. Teams responsible for coordinating modifications or new developments must now also manage configurations that are part of and have implications for enterprise capabilities like networking, information assurance, and data management. This requires new perspectives on what needs to be managed at what level (system or enterprise) and with what stakeholder involvement. Enterprise CM activities need to be appropriately resourced to meet the greater needs of an enterprise. For more information on enterprise systems, see the SEG's Enterprise Engineering section.

Sustaining and maintaining organizations are critical CM stakeholders. Make sure the sustainment or maintenance organization is on board and an equal partner in the CM process. Their unique vantage point helps ensure the sustainability of the fielded baseline by identifying vanishing vender items, provisioning, and a host of other logistical considerations.

References and Resources

1. Acquisition Community Connection, Defense Acquisition Guidebook, 4.2.3.1.6, accessed January 25, 2010.

Additional References and Resources

Air Force Instruction 63-131, November 6, 2009, *Modification Program Management.*

Leffingwell, D. and D. Widrig, 2003, "Managing Change," *Managing Software Requirements: A Use Case Approach*, Addison-Wesley Professional, Chapter 28.

Definition: *Webster defines a tool as "something regarded as necessary to the performance of one's occupation or professional task. [Words are the tools of my trade.]" Configuration Management (CM) tools come in several forms. For the systems engineers and their partners/sponsors/customers, these tools include best practice methodologies, standards, documentation, managed environments, manual tools, automated tools, and leadership skills. These require and enable discipline and rigor needed to plan, stand up, implement, and carry out CM successfully.*

Keywords: *automated tools, configuration management policy, program management plan, Statement of Work (SOW), tools*

CONFIGURATION MANAGEMENT

Configuration Management Tools

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to have a sound understanding of the principles of configuration management; how program management and project management view configuration management; how the development organization initiates configuration control; how the developer implements configuration management; and how the government sponsor or sustainment organization continues the configuration management following product delivery. MITRE SEs are generally involved in identifying requirements for automated tools to support the CM process. Rather than selecting specific automated CM tools, MITRE SEs need to begin with requirements that understand and address the roles of technical and non-technical elements of CM, to include documentation and the traditional software configuration management elements of hardware and software. To be successful, it is essential to understand CM. CM success is a function of leadership to insist on its implementation and use.

CM is defined in the SEG's Configuration Management topic. Within that context, it is important to note that CM is not all things to all people, nor is it program management. It is a tool used by program management. It is not document management, but it is a partner tool used by document management. It is not requirements management nor engineering, but is a tool with important connections to requirements engineering activities and processes.

Automated CM tools can help:

- Record, control, and correlate Configuration Items (CIs), Configuration Units (CUs), and Configuration Components (CCs) within a number of individual baselines across the life cycle.
- Identify and control baselines.
- Track, control, manage, and report change requests for the baseline CIs, CCs, and CUs.
- Track requirements from specification to testing.
- Identify and control software versions.
- Track hardware parts.
- Enable rigorous compliance with a robust CM process.
- Conduct Physical Configuration Audits (PCAs).
- Facilitate conduct of Functional Configuration Audits.

Best Practices and Lessons Learned

Start at the beginning. Get top-down buy-in on the value of CM. A successful CM program is supported and enforced by leadership. Set or cite a CM policy/directive that authorizes a high-level Configuration Control Board (CCB) (e.g., Executive CCB) at the highest authority with links to higher and lower boards. Coordinate CM planning with requirements development and management, quality assurance, process improvement, independent validation and verification, and existing enterprise processing centers to ensure engagement and integration of production stakeholders. Ensure the Program Management Plan contains a program-level Configuration Management Plan. Coordinate with the acquisition organization (e.g., Contract Officer, Contract Officer Technical Representative Acquisition Advisor) to ensure adequate CM requirements are in the SOW. In addition to the standard CM requirements, the SOW should include formal CM audits of the contractors to measure compliance with the agency/customer CM policy, agency/customer regulations, etc.

Audit early and often. Set standards early, and audit for compliance. Identify or establish agency/government/enterprise policy, plan, practice, procedures, and standards, including naming and tagging conventions early. Audit internally to ensure the Program Management Office is following the policies and procedures. Consider an annual demonstration of contractor alignment with Software Engineering Institute Capability Maturity Model Integrated (CMMI) CM, Information Technology Infrastructure Library CM, uniform top-level CM processes—ISO-9001, and National Consensus Standard for Configuration Management (ANSI EIA 649). Consider using a CMMI Practice Implementation Indicator Description format, such as those used for CMMI assessments, and include conclusive evidence for the demonstration. Schedule it annually.

Considerations for automated CM tool acquisition. First, ascertain the method of management that is most significant for your project or system, and ensure the tools serve that purpose. Next, define the requirements. Automated CM tool requirements need to be identified before acquisition decisions are made. It is critical to establish requirements for the automated CM tool by collecting available CM plans, policy, process, procedure, and instructions, and meeting with the relevant stakeholders. Be certain to include business, user, contractor, and operations and maintenance stakeholders to define the automated CM tool requirements.

These requirements should include considerations for the following:

- Requirements Management
- Document Management version control
- Controlled repositories
- Configuration Identification and control, including hardware, software, etc.
- Change Request processing and tracking
- Audit support
- Configuration Status Accounting Reporting (CSAR)
- Baseline management (software, documentation, requirements, design, product, production)
- Software development check-out/check-in
- CM of environments (development, test, product, production); may be multiples of each
- Multiplatform capabilities (personal computer, local area network, Web, mainframe, data centers, etc.)
- Release engineering of all types of Change Requests (CRs) (e.g., normal releases, routine releases such as operating system and security updates and patches, break fixes, emergency)
- Transition CM tools into the sustainment activities
- Automated CM tool within the approved technical reference model or fully justifiable for a waiver.

CM tool selection needs to include a discussion of the selection criteria based on the requirements, evaluation of tools, and selection of tools.

CM Lessons Learned and Pitfalls

- Depending on the level of support from the program leadership and stakeholders for CM, tools may not be included as part of the overall CM plan or planned for acquisition. If automated tools are acquired, ensure that program leadership is aware of the need for planning short-, mid-, and long-term needs for installation, establishing the baseline data, and training, updating, securing, and maintaining the tools and the associated process and procedures needed to use them effectively.
- Set expectations early. CM and configuration change control are all about CRs, regardless of what they are called, and the impact of change on scope, cost, and schedule.
- Keep informal and formal communication open with CM as an agenda topic in meetings and gate reviews. Do not shoot the messengers.
- Consider release management separate from CM. Do not assume that release management can be done by the CM organization.
- Everyone may know CM, but training will be needed to orient staff on how CM is done in your particular organization.
- Coordinate with those responsible for business continuity of operation and disaster recovery. Some will assume that CM will provide the capability to restore the entire system if the need arises. This is not a safe assumption, unless your CM tools are designed with this capability in mind.
- Identify, establish, maintain, and control the necessary development, test, and production environments, including the automated CM tools, hardware, software, operating system, security, access control, and supportive infrastructure.
- Contractors and periods of performance may come and go. It is recommended that the transition from one contractor to the next include an inventory of baselined hardware, software, documents, etc. PCAs on the departing contractor product should establish what the new contractor inherits. Gap analysis should be performed to determine the delta and provide input to the contract closure activity prior to making final payments to the departing contractor.

Conversely, if the above lessons are not applied, the consequences can lead to CM failure. Indications that things are not going well include: leadership support is not evident, formal CCBs are not chartered or recognized as change approval authorities or do not function, “lanes in the road” are not defined and chaos reigns, attendance at CM meetings (CCBs, engineering/technical review boards, and impact assessment boards/teams) declines or is non-existent, and cross program/project impacts are not identified by CM, but only when something breaks down.

Automated CM Tools Lessons Learned

- Buying a tool will not establish an appropriate CM program for your effort.
- It is unlikely that a single automated CM tool can be all things to all stakeholders by integrating all required elements across all platforms. So-called commercial off-the-shelf "suites" of tools may not contain integrated capabilities to suit the enterprise. When they have the potential for integration, often there may be a significant effort needed to adapt the products after they are taken out of the box. What may support software development with check-out/check-in features may be bundled within a "suite" of stand-alone automated tools without any integration. Tool administrators may only have the ability to export to a spreadsheet for reporting. Automated tools may control one area well, but not be suited for other areas.
- The automated CM tools used within the development and test environments may not be compatible with those in the production environments. This may require development of semi-automated or manual processes. In either event, security and firewall infrastructures may present additional challenges.
- Automated CM tools may offer flexible options, including the ability to design your own change request (CR) form and flow. This has inherent pros and cons. There is often an assumption by the acquirer that the tool will deliver a CR process out of the box such that no other development effort will be needed. It is important to understand that the capabilities delivered out of the box are directly impacted by the installation/customization of those tools. It is important to understand the need for development and administration of the automated tools, and set expectations early on.
- When planning the acquisition of the automated CM tool, consider the initial and longer term costs, including licenses, and labor cost to install and develop it so it is usable for your program. Plan for ongoing system administration, security, maintenance, backup, and recovery as well as business continuity of operation and disaster recovery. Consider the ability of the tool, and data contained within the tool, to be transitioned from one contractor to another, which is sometimes the case when a program transitions from production to sustainment.
- Avoid an approach with tools that implies the tool will establish the standard and solve the problem. It is *not* best practice to say "Here is the solution. What is your problem?"

References and Resources

MITRE Center for Connected Government, CM Toolbox.

MITRE Systems Engineering Practice Office, Configuration Management Toolkit.

Integrated Logistics Support

Definition: *Integrated logistics support (ILS) is the management and technical process through which supportability and logistic support considerations are integrated into the design of a system or equipment and taken into account throughout its life cycle. It is the process by which all elements of logistic support are planned, acquired, tested, and provided in a timely and cost-effective manner [1].*

Keywords: *acquisition logistics, computer resources support, design interface, life-cycle cost, life-cycle logistics, maintenance planning, technical data*

Context

Supportability and life-cycle costs are generally driven by technical design that occurs during the development stage of the acquisition process. The experience of the U.S. Department of Defense (DoD) has consistently shown that the last 5 percent increase in performance adds a disproportionate increase in life-cycle costs, generally due to the use of new and often unproven technologies and design, which drive up supportability costs. This increase in costs is found regardless of the support plan (organic or contractor support) and is a significant cost factor in the total sustainment budget of government departments and agencies.

MITRE SE Roles and Expectations

MITRE staff are expected to understand the impact of technical decisions made during design and development on the usability and life-cycle support of the system. They are expected to account for life-cycle logistics considerations as part of the systems engineering process.

Introduction

Traditionally, MITRE systems engineering support to acquisition programs has not strongly focused on ILS. However, life-cycle costs are increasingly driven by engineering considerations during the early stages of acquisition. As a consequence, there is an important cause-and-effect relationship between systems engineering and ILS that systems engineers need to be aware of and account for in their activities.

In addition to the general ILS best practices and lessons learned discussed below, this topic contains two articles. The article "Reliability, Availability, and Maintainability" discusses best practices and lessons learned in developing design attributes that have significant impacts on the sustainment or total Life Cycle Costs (LCC) of a system. A companion article, "Affordability, Efficiency, and Effectiveness (AEE)," in the Systems Engineering Life-Cycle Building Blocks section also contains best practices and lessons learned for managing LCC. The second article under this topic is an ILS subject of special interest. The article "Managing Energy Efficiency" discusses engineering considerations of conserving energy resources during production, operations, and disposal of systems. Impacts include not just environmental concerns but also system performance parameters (e.g., range of host vehicle) and recurring operating costs. For example, engineering considerations of data centers are increasingly focused on technical issues and cost concerns of the energy needed to run them.

Best Practices and Lessons Learned

The computer resources working group. Computer resources support encompasses the facilities, hardware, software, documentation, manpower, and personnel needed to operate and support mission-critical computer hardware/software systems. As the primary end item, support equipment, and training devices all increase in complexity, more and more software is being used. The expense associated with the design and maintenance of software programs is so high that no one can afford to poorly manage this process. It is critical to establish some form of computer resource working group to accomplish the necessary planning and management of computer resources support [2].

The impact of maintenance planning. Maintenance planning establishes maintenance concepts and requirements for the life of the system. It includes, but is not limited to:

- Levels of repair
- Repair times

- Testability requirements
    - Support equipment needs
    - Manpower and skills required
    - Facilities
    - Interservice, organic, and contractor mix of repair responsibility
    - Site activation.

This element has a great impact on the planning, development, and acquisition of other logistics support elements. Taken together, these items constitute a substantial portion of the recurring cost (and therefore life-cycle cost) of a procurement. Another factor related to this, and one that should be seriously considered, is energy use and efficiency. The rising cost of energy and its proportion within the overall recurring cost should be managed proactively (refer to the article on "Managing Energy Efficiency" within this section of the Guide).

Early consideration of manpower requirements. Manpower and personnel involves the identification and acquisition of personnel (military and civilian) who have the skills and grades required to operate, maintain, and support systems over their lifetime. Early identification is essential. If the needed manpower requires adding staff to an organization, a formalized process of identification and justification needs to be made to higher authority. Add to this the necessity to train these staff, new and existing, in their respective functions on the new system, and the seriousness of any delay in accomplishing this element becomes apparent. In the case of user requirements, manpower needs can, and in many cases do, ripple all the way back to recruiting quotas. Required maintenance skills should be considered during the design phase of the program; unique technology often requires unique skills for maintenance. Note that information technology expertise and information security skills can still be lacking in current organic maintenance resources and may require investment in adequate training.

Ensuring a supply support structure. Supply support consists of all the management actions, procedures, and techniques necessary to determine requirements to acquire, catalog, receive, store, transfer, issue, and dispose of spares, repair parts, and supplies (including energy sources and waste). In lay terms, this means having the right spares, repair parts, and supplies available, in the right quantities, at the right place, right time, and right price. The process includes provisioning for initial support as well as acquiring, distributing, and replenishing inventories. An aircraft can be grounded just as quickly for not having the oil to put in the engine as it can for not having the engine.

Evaluating the supply chain. As stated above, access to spare equipment and supplies is critical to the operation of the system delivered. Not only should there be a logistics process in place to handle spares and supplies, there also needs to be assurance that the supply chain will continue or have alternate sources. Care should be taken to assess supply chains for continued viability: avoidance of diminishing manufacturing supply, identification of alternatives, and mission assurance. An article "Supply Chain Risk Management" is in the SEG's Enterprise Engineering section. (To learn more about mission assurance, read the article "Cyber Mission Assurance" in the SEG's Enterprise Engineering section.)

Minimizing unique support requirements. Ensure all the equipment (mobile or fixed, hardware or software) required to support the operation and maintenance of a system is identified and provided. This includes ground handling and maintenance equipment, tools, metrology and calibration equipment, manual and automatic test equipment, modeling and simulation used for testing, and any software debugging/monitoring applications necessary for software maintenance or modification. Acquisition programs should look to decrease the proliferation of unique support equipment into the inventory by minimizing the development of new support equipment and giving more attention to the use of existing government or commercial equipment.

The benefits of pre-test training. A training plan and program should consist of all policy, processes, procedures, techniques, training devices, and equipment used to train all user personnel to acquire, operate, and support a system. This includes individual and crew training; new equipment training; and initial, formal, and on-the-job training. Although the greatest amount of training is accomplished just prior to fielding a system, in most programs a large number of individuals must also be trained during system development to support the system test and evaluation program. This early training also provides a good opportunity to flush out all training issues prior to the production and conduct of formal training.

Design interface. This is the relationship of logistics-related design parameters to readiness and support resource requirements. Logistics-related design parameters include:

- Reliability and maintainability
- Human factors
- System safety
- Survivability and vulnerability
- Hazardous material management
- Standardization and interoperability
- Energy management/efficiency
- Corrosion
- Nondestructive inspection
- Transportability.

These logistics-related design parameters are expressed in operational terms rather than as inherent values, and specifically relate to system readiness objectives and support costs. Design interface really boils down to evaluating all facets of an acquisition, from design to support and operational concepts for logistical impacts, to the system itself and the logistics infrastructure.

The importance of technical data. The term "technical data" represents recorded information of a scientific or technical nature, regardless of form or character (such as manuals and drawings). Computer programs and related software are not technical data; documentation of computer programs and related software is. Technical manuals and engineering drawings are the most expensive and probably the most important data acquisitions made in support of a system, because they provide the instructions for its operation and maintenance. Generation and delivery of technical data should be specified early on in the acquisition planning process (within requests for proposals and statements of work, etc.) to ensure consideration and cost. Absence of technical data adversely impacts the operations and maintenance of a system and may result in a sole-source situation for the developer.

More on Technical Data

Since July 2006, a number of important Department of Defense (DoD) developments related to technical data rights have transpired, including:

- Issuance of Secretary of the Air Force Memo, May 3, 2006, "Data Rights and Acquisition Strategy." [3]
- Issuance of GAO Report "Weapons Acquisition: DoD Should Strengthen Policies for Assessing Technical Data Needs to Support Weapon Systems, July 2006." [4]
- Passage of Congressional Language in PL 109-364, FY07 John Warner National Defense Authorization Act. [5]
- Issuance of USD AT&L Policy Memo "Data Management and Technical Data Rights," [6] July 19, 2007, which requires program managers to assess long-term technical data requirements for all ACAT I and II programs, regardless of the planned sustainment approach, and reflect that assessment in a data management strategy (DMS).
- Data Management and Technical Data Rights wording will be added to the next update of the DoD Instruction 5000.2. [7]
- Issuance of DFARS Interim Rule, September 6, 2007. [8]
- Issuance of US Army ASA (ALT) policy memorandum "Data Management and Technical Data Rights, April 1, 2008." [9]

To encourage creative and well–thought-out development of data management strategies (DMS) (which may well include access rather than procurement of data), the DoD has chosen not to issue a standard DMS format/template. The cost of actual ownership of the data, including storage, maintenance, and revision, is high and DoD has found it more economical for the provider to manage the data as a service through the life cycle of the system. However, according to Office of the Secretary of Defense staff, at a minimum a DMS should address:

- Specific data items required to be managed throughout the program's life cycle
- Design, manufacture, and sustainment of the system
- Re-compete process for production, sustainment, or upgrade
- Program's approach to managing data during acquisition and sustainment (i.e., access, delivery, format)
- Contracting strategy for technical data and intellectual property rights
- Any requirements/need for a priced option
- Any unique circumstances.

See also the Interactive Electronic Technical Data (IETM) site [10] and the DAU Data Management (DM) Community of Practice (CoP) [11] for further information.

References and Resources

1. U.S. Department of Defense, 2005, Dictionary of Military and Associated Terms.
2. "Computer Resources Support," Acquisition Community Connection, accessed February 4, 2010.
3. May 3, 2006 Secretary of the Air Force Memo, "Data Rights and Acquisition Strategy."
4. GAO Report, July 2006, "Weapons Acquisition: DOD Should Strengthen Policies for Assessing Technical Data Needs to Support Weapon Systems."
5. PL 109-364, FY07 John Warner National Defense Authorization Act.
6. USD AT&L, July 19, 2007, Policy Memo, "Data Management and Technical Rights."
7. DoD Instruction 5000.2.
8. DFARS Interim Rule Issued September 6, 2007.
9. US Army ASA (ALT) policy memorandum, April 1, 2008, "Data Management and Technical Data Rights."
10. IETM, Interactive Electronic Technical Manuals, https://acc.dau.mil/ietm
11. DAU Data Management (DM) Community of Practice (CoP), https://acc.dau.mil/CommunityBrowser.aspx?id=17647

Additional References and Resources

Assistant Secretary of Defense for Logistics and Materiel Readiness, Logistics and Materiel Readiness website, accessed February 4, 2010.

Defense Acquisition University, accessed February 4, 2010.

Definition: *Reliability, availability, and maintainability (RAM or RMA) are system design attributes that have significant impacts on the sustainment or total life cycle costs (LCC) of a developed system. RAM attributes impact the ability to perform the intended mission and affect overall mission success. Reliability is typically defined as the probability of zero failures over a defined time interval (or mission), whereas availability is the percentage of time a system is considered ready to use when tasked. Maintainability is a measure of the ease and rapidity with which a system or equipment can be restored to operational status following a failure.*

Keywords: *availability, maintainability, RAM, reliability, RMA*

INTEGRATED LOGISTICS SUPPORT

Reliability, Availability, and Maintainability

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to understand the purpose and role of reliability, availability, and maintainability (RAM) in the acquisition process, where it occurs in systems development, and the benefits of employing it. MITRE SEs are also expected to understand and recommend when RAM is appropriate to a situation and if the process can be tailored to meet program needs. They are expected to understand the technical requirements for RAM as well as strategies and processes that encourage and facilitate active participation of end users and other stakeholders in the RAM process. They are expected to monitor and evaluate contractor RAM technical efforts and the acquisition program's overall RAM processes and recommend changes when warranted.

Background

Reliability is the wellspring for the other RAM system attributes of Availability and Maintainability. Reliability was first practiced in the early start-up days for the National Aeronautics and Space Administration (NASA) when Robert Lusser, working with Dr. Wernher von Braun's rocketry program, developed what is known as "Lusser's Law" [1]. Lusser's Law states that that the reliability of any system is equal to the product of the reliability of its components, or the so-called weakest link concept.

The term "reliability" is often used as an overarching concept that includes availability and maintainability. Reliability in its purest form is more concerned with the probability of a failure occurring over a specified time interval, whereas availability is a measure of something being in a state (mission capable) ready to be tasked (i.e., available). Maintainability is the parameter concerned with how the system in use can be restored after a failure, while also considering concepts like preventive maintenance and Built-In-Test (BIT), required maintainer skill level, and support equipment. When dealing with the availability requirement, the maintainability requirement must also be invoked as some level of repair and restoration to a mission-capable state must be included. One can see how logistics and logistic support strategies would also be closely related and be dependent variables at play in the availability requirement. This would take the form of sparing strategies, maintainer training, maintenance manuals, and identification of required support equipment. The linkage of RAM requirements and the dependencies associated with logistics support illustrates how the RAM requirements have a direct impact on sustainment and overall LCC. In simple terms, RAM requirements are considered the upper level overarching requirements that are specified at the overall system level. It is often necessary to decompose these upper level requirements into lower level design-related quantitative requirements such as Mean Time Between Failure/Critical Failure (MTBF or MTBCF) and Mean Time To Repair (MTTR). These lower level requirements are specified at the system level; however, they can be allocated to subsystems and assemblies. The most common allocation is made to the Line Replaceable Unit (LRU), which is the unit that has lowest level of repair at the field (often called organic) level of maintenance.

Much of this discussion has focused on hardware, but the complex systems used today are integrated solutions consisting of hardware and software. Because software performance affects the system RAM performance requirements, software must be addressed in the overall RAM requirements for the system. The wear or accumulated stress mechanisms that characterize hardware failures do not cause software failures. Instead, software exhibits behaviors that operators perceive as a failure. It is critical that users, program offices, test community, and contractors agree early as to what constitutes a software failure. For example, software "malfunctions" are often recoverable with a reboot, and the time for reboot may be bounded before a software failure is declared. Another issue to consider is frequency of occurrence even if the software reboot recovers within the defined time window, as this will give an indication of stability of the software. User perception of what constitutes a software failure will surely be influenced by both the need to reboot and the frequency of "glitches" in the operating software.

One approach to assessing software "fitness" is to use a comprehensive model to determine the current readiness of the software (at shipment) to meet customer requirements. Such a model needs to address quantitative parameters (not just process elements). In addition, the method should organize and streamline existing quality and reliability data into a simple metric and visualization that are applicable across products and releases. A novel, quantitative software readiness criteria model [2] has recently been developed to support objective and effective decision making at product shipment. The model has been "socialized" in various forums and is being introduced to MITRE work programs for consideration and use on contractor software development processes for assessing maturity. The model offers:

- An easy-to-understand composite index
- The ability to set quantitative "pass" criteria from product requirements
- Easy calculation from existing data
- A meaningful, insightful visualization
- Release-to-release comparisons
- Product-to-product comparisons
- A complete solution, incorporating almost all aspects of software development activities

Using this approach with development test data can measure the growth or maturity of a software system along the following five dimensions:

- Software Functionality
- Operational Quality
- Known Remaining Defects (defect density)
- Testing Scope and Stability
- Reliability

For greater detail, see ref. [2].

Government Interest and Use

Many U.S. government acquisition programs have recently put greater emphasis on reliability. The Defense Science Board (DSB) performed a study on Developmental Test and Evaluation (DT&E) in May 2008 and published findings [3] that linked test suitability failures to a lack of a disciplined systems engineering approach that included reliability engineering. The Department of Defense (DoD) has been the initial proponent of systematic policy changes to address these findings, but similar emphasis has been seen in the Department of Homeland

Security (DHS) as many government agencies leverage DoD policies and processes in the execution of their acquisition programs.

As evidenced above, the strongest and most recent government support for increased focus on reliability comes from the DoD, which now requires most programs to integrate reliability engineering with the systems engineering process and to institute reliability growth as part of the design and development phase [4]. The scope of reliability involvement is further expanded by directing that reliability be addressed during the Analysis of Alternatives (AoA) process to map reliability impacts to system LCC outcomes [5]. The strongest policy directives have come from the Chairman of the Joint Chiefs of Staff (CJCS) where a RAM-related sustainment Key Performance Parameter (KPP) and supporting Key System Attributes (KSAs) have been mandated for most DoD programs [6]. Elevation of these RAM requirements to a KPP and supporting KSAs will bring greater focus and oversight, with programs not meeting these requirements prone to reassessment and reevaluation and program modification.

Best Practices and Lessons Learned [7] [8]

Subject matter expertise matters. Acquisition program offices that employ RAM subject matter experts (SMEs) tend to produce more consistent RAM requirements and better oversight of contractor RAM processes and activities. The MITRE systems engineer has the opportunity to "reach back" to bring MITRE to bear by strategically engaging MITRE-based RAM SMEs early in programs.

Consistent RAM requirements. The upper level RAM requirements should be consistent with the lower level RAM input variables, which are typically design related and called out in technical and performance specifications. A review of user requirements and flow down of requirements to a contractual specification document released with a Request For Proposal (RFP) package must be completed. If requirements are inconsistent or unrealistic, the program is placed at risk for RAM performance before contract award.

Ensure persistent, active engagement of all stakeholders. RAM is not a stand-alone specialty called on to answer the mail in a crisis, but rather a key participant in the acquisition process. The RAM discipline should be involved early in the trade studies where performance, cost, and RAM should be part of any trade-space activity. The RAM SME needs to be part of requirements development with the user that draws on a defined Concept of Operations (CONOPS) and what realistic RAM goals can be established for the program. The RAM SME must be a core member of several Integrated Product Teams (IPTs) during system design and development to establish insight and a collaborative relationship with the contractor team(s): RAM IPT, Systems Engineering IPT, and Logistics Support IPT. Additionally, the RAM specialty should be part of the test and evaluation IPT to address RAM test strategies (Reliability Growth, Qualification tests, Environmental testing, BIT testing, and

Maintainability Demonstrations) while interfacing with the contractor test teams and the government operational test community.

Remember—RAM is a risk reduction activity. RAM activities and engineering processes are a risk mitigation activity used to ensure that performance needs are achieved for mission success and that the LCC are bounded and predictable. A system that performs as required can be employed per the CONOPS, and sustainment costs can be budgeted with a low risk of cost overruns. Establish reliability Technical Performance Measures (TPMs) that are reported on during Program Management Reviews (PMRs) throughout the design, development, and test phases of the program, and use these TPMs to manage risk and mitigation activities.

Institute the Reliability Program Plan. The Reliability (or RAM) Program Plan (RAMPP) is used to define the scope of RAM processes and activities to be used during the program. A program office RAMPP can be developed to help guide the contractor RAM process. The program-level RAMPP will form the basis for the detailed contractor RAMPP, which ties RAM activities and deliverables to the Integrated Master Schedule (IMS).

Employ reliability prediction and modeling. Use reliability prediction and modeling to assess the risk in meeting RAM requirements early in the program when a hardware/software architecture is formulated. Augment and refine the model later in the acquisition cycle, with design and test data during those program phases.

Reliability testing. Be creative and use any test phase to gather data on reliability performance. Ensure that the contractor has planned for a Failure Review Board (FRB) and uses a robust Failure Reporting And Corrective Action System (FRACAS). When planning a reliability growth test, realize that the actual calendar time will be 50–100% more than the actual test time to allow for root cause analysis and corrective action on discovered failure modes.

Don't forget the maintainability part of RAM. Use maintainability analysis to assess the design for ease of maintenance, and collaborate with Human Factors Engineering (HFE) SMEs to assess impacts to maintainers. Engage with the Integrated Logistics Support (ILS) IPT to help craft the maintenance strategy, and discuss levels of repair and sparing. Look for opportunities to gather maintainability and testability data during all test phases. Look at Fault Detection and Fault Isolation (FD/FI) coverage and impact on repair time lines. Also consider and address software maintenance activity in the field as patches, upgrades, and new software revisions are deployed. Be aware that the ability to maintain the software depends on the maintainer's software and IT skill set and on the capability built into the maintenance facility for software performance monitoring tools. A complete maintenance picture includes defining scheduled maintenance tasks (preventive maintenance) and assessing impacts to system availability.

Understand reliability implications when using COTS. Understand the operational environment and the COTS hardware design envelopes and impact on reliability performance. Use Failure Modes Effects Analysis (FMEA) techniques to assess integration risk and characterize system behavior during failure events.

References and Resources

1. Military Handbook 338, Electronic Reliability Design Handbook, October 1998.
2. Asthana, A., and J. Olivieri, Quantifying Software Reliability and Readiness, *IEEE Communications Quality Reliability (CQR) Proceedings*, 2009.
3. Report of the Defense Science Board Task Force on Developmental Test and Evaluation, May 2008.
4. Department of Defense, December 2008, Instruction Number 5000.02, Operation of the Defense Acquisition System.
5. Department of Defense, June 2009, *Reliability, Availability, Maintainability, and Cost Rationale Report Manual*
6. Department of Defense, January 2011, *Manual for the Operation of the Joint Capabilities Integration and Development System*.
7. Department of Defense, August 2005, DoD Guide for Achieving Reliability, Availability, and Maintainability.
8. Reliability Information Analysis Center, Reliability Toolkit: Commercial Practices Edition.

Definition: *Energy efficiency is a measure of how well a system uses the available energy potential of its inputs. The goal of managing it is about using less energy to provide the same level of service, thus conserving our energy resources during production, operations, and disposal of systems. The impact goes beyond environmental concerns; it also has implications on the range of vehicles, the cost of operations, and the future of a system.*

Keywords: *efficiency, energy, green, life-cycle costs, trade-off*

INTEGRATED LOGISTICS SUPPORT

Managing Energy Efficiency

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to be able to understand the impact of technical decisions on the energy efficiency of a system. They are expected to account for energy efficiency considerations across the system life cycle as part of the systems engineering process.

Background and Motivation

Many engineers have studied efficiency in the past and are familiar with the theoretical limits of a given thermodynamic cycle. Increased attention to energy efficiency has come about in the last 50 years, primarily as a result of a continuing trend of energy crises throughout the world. The first modern major crisis was in 1973. The oil crisis resulted not from a depletion of energy sources, but from political maneuvers that took the form of the Arab Oil Embargo. The effects were sharply felt throughout the country, and resulted in the price of oil quadrupling [1], rationing, and long lines. Other crises occurred in 1979, associated with the Iranian Revolution, and in 1990, over the Iraqi invasion of Kuwait. Several times during this century, price increases in oil were caused by reports of declining petroleum reserves, natural disasters such as Hurricane Katrina, political activities in other countries, and conflicts that did not directly involve the United States [2, 3]. Each of these demonstrates the importance of considering all elements of POET (political, operational, economic, technical), or as expanded, TEAPOT (technically accurate, economically feasible, actionable, politically and culturally insightful, operationally grounded, and timely) [4].

Complicating the landscape is the fact that most energy today is produced by non-renewable resources in the form of coal, fissile materials, petroleum, and natural gas. When production first begins with each material, the amount of material recovered for each unit of energy is large. For example, in the mid-nineteenth century when oil production began, the largest oil fields could recover 50 barrels of oil for every barrel used in extraction, refining, and transportation. Today that number is between one and five. Once it reaches one (one barrel of oil to make one barrel of oil), the only way to make the resource available for consumption is to use other energy sources to bring it to market. This situation will happen before the resource is physically exhausted. Energy resource production follows nearly a bell-shaped curve called the Hubbert curve [5], and trails discovery by about 35 years. In the United States, peak oil (the peak of the Hubbert curve) was reached in the 1970s and has resulted in the country importing increasing amounts as the consumption continued to increase since that time. For the world, peak oil is estimated to be 2020. As production falls, the production process gets more expensive and the price rises. The same happens for each natural resource. Understanding this sequence is important to understanding the push for increased energy efficiency.

The number of U.S. military bases overseas has declined over the past two decades, and it is getting increasingly difficult to gain access to countries to forward deploy forces and equipment. This trend and the desire for the U.S. military to minimize the risk to its forces have resulted in programs that deliver weapons systems that operate from longer ranges. To achieve this, either an increased infrastructure of resupply systems must be put in place, or weapons systems must be more efficient to achieve the longer ranges needed to accomplish the missions. As for the resupply chain, there is also a desire to reduce the forward footprint of deployed forces, including generators and the resupplied fuels (an issue now with complex weapon and computing systems and their power consumption).

Government Interest and Use

Energy efficiency is a parameter that can potentially be adjusted during the development of a system. It has a direct effect on the range of vehicles, as well as the upfront and recurring operating costs of systems. Secondary effects include minimum and maximum operating limits, and adverse effects on performance parameters like speed and dynamic control. In some domains, increased energy efficiency generally means higher upfront costs with the savings realized over the life cycle in the form of decreased energy bills.

The government uses energy to run its transportation fleets, surveillance and weapons delivery systems, facilities, and information processing systems. Trends in data centers have shifted the predominate cost from the building itself to the operation of the building and, more specifically, to obtaining the energy. The Green Grid [6, 7] has established a simple set of metrics that can be used to reduce the energy consumption for the data center. While they are very simplistic, such as the ratio of power used by IT equipment to the power entering the facility, they give an indication of overall data center energy efficiency. Many organizations have used these measurements as the tool to reduce the cost of operations. These include Google, Mass Mutual, Patagonia, and Kimberly-Clark [8, 9, 10, 11, 12]. To date, more work in this area is being done in the commercial arena than in the government sector.

The government has established green standards for acquisitions, part of which mandate the use of Energy Star™ and EPEAT™ [13, 14, 15]. There has been an increased use of the U.S. Green Building Council LEED™ program in the acquisition of buildings. It is important to note that these buildings must be maintained and consistently improved to keep the energy efficiency of the building competitive. As with all green initiatives, the target continues to move over time [16, 17].

Best Practices and Lessons Learned

Be wary of advertised efficiency claims. Programs such as Energy Star™, EPEAT™, LEED™, and Green Globes (international counterpart to LEED™) need to be carefully scrutinized for how they gauge energy consumption. For example, a printer rated by Energy Star™ can be expected to achieve the stated energy efficiency only when operated in exactly the same manner in which the rating was obtained. Generally the test conditions used are not provided. So, there is the need to be aware of or find out what is being measured and how it applies to your system's situation [13, 14, 18, 19, 20, 21].

Know your system's operations. Thoroughly understand how the government intends to use the system before evaluating its energy consumption profile and requirements. Understand the level of energy management expertise in the contractor and the maintenance organization because, although components of a system may have energy-saving modes, they may be shipped in a configuration that does not enable the modes. Also, look into whether operational, information security, or other policies or procedures prevent or impede their use [22]. As an example, in many organizations it is common practice to push out software updates during the early morning hours. If a system is in a low power mode, it may not respond to the push. To avoid this situation, an organization may have policies that prevent systems from entering a low power mode.

Take measures. The only way to truly know what the system is doing is to measure its performance. The more data you have, the greater your ability to make informed decisions about your system's operation, whether it be a weapons system, a building, or a data center. With more knowledge, you are better able to know where and how energy is being used and recommend solutions to improve energy efficiency [16, 23, 24].

Involve all stakeholders. Energy efficiency is an enterprise-level problem. In general, the organization paying for the energy is not the organization procuring the system or facility, nor the organization using or making decisions about using the energy. Be sure to involve all stakeholders [9, 17].

References and Resources

1. CBC News in Depth, July 18, 2007, The Price of Oil, Marching to $100?, CBC News.
2. Cooper P. J., July 8, 2006, "Record Oil Price sets the scene for $200 next year," *AMEinfo.com*.
3. Mortished, Carl, August 30, 2005, "Hurricane Katrina whips oil price to a record high," *The Times*.
4. The MITRE Corporation, CCG Quality Program, CEM Quality Handbook and Quality Tools.
5. Grove, N., June 1974, "Oil, the Dwindling Treasure," *National Geographic*.
6. "Guidelines for Energy-Efficient Datacenters," The Green Grid, February 16, 2007.
7. The Green Grid, http://www.thegreengrid.org/, accessed February 17, 2014.
8. Anderson, S. F., November 2009, "Improving Data Center Efficiency," *World Energy Engineering Congress (WEEC)*.
9. Belady, C., September 5, 2006, "How to Minimize Data Center Utility Bills," *E-Business News*.

10. Dixon, S., November 2009, "Energy Management for Commercial Office Buildings: An Owner's Perspective," *WEEC*.
11. McMahon, J., November 3, 2009, "Sustainable Distribution," *Sustainable Facility*.
12. Marklein, B. Richard, and J. Marin, November 2009, "Kimberly-Clark's Energy Management for 2015," *WEEC*.
13. Bush, G. W., January 24, 2007, Executive Order 13423, Strengthening Federal Environmental, Energy, and Transportation Management.
14. Energy Star, http://www.energystar.gov/, accessed February 17, 2014.
15. EPEAT, http://www.epeat.net/, accessed February 14, 2014.
16. Huppes, G., and M. Ishikawa, October 2005, "A Framework for Quantified Eco-efficiency Analysis," *Journal of Industrial Ecology,* vol. 9 no. 4, pp. 25–41.
17. Myatt, B., September/October 2009, "Low-Cost Data Center Energy Efficiency Programs in Action," *Mission Critical*, pp. 20–21.
18. Boyd, Gale A., July 2005, "A Method for Measuring the Efficiency Gap between Average and Best Practice Energy Use, The ENERGY STAR Performance Indicator," *Journal of Industrial Ecology*, vol. 9 no. 3, pp. 51–65.
19. DOE and EPA, Fuel Economy, (MPG and other information about vehicles), accessed February 26, 2010.
20. Green Building Initiative, Green Globes: the practical building rating system, accessed February 26, 2010.
21. U.S. Green Building Council (LEED) website, accessed February 26, 2010.
22. Nordman, Bruce, Mary Ann Peitte, Kris Kinney, and Carrie Webber, January 1997, User Guide to Power Management in PCs and Monitors, Lawrence Berkeley National Laboratory.
23. 80 PLUS Energy-Efficient Technology Solutions, http://www.plugloadsolutions.com/80PlusPowerSupplies.aspx, accessed February 17, 2014.
24. Group, J., November 2009, "Success Stories in Cutting Facility Energy Costs Using Electric Submeters," *WEEC*.

Additional References and Resources

Rabiee, A., H. Shayanfar, and N. Amjady, Jan-Feb 2009, "Reactive Power Pricing," *IEEE Power and Energy Magazine*, vol. 7, issue 1, pp. 18–32.

The MITRE Corporation, "Integrated Logistics Support," MITRE Systems Engineering Competency Model, accessed February 26, 2010.

Quality Assurance and Measurement

Definition: *Quality assurance is "a planned and systematic means for assuring management that the defined standards, practices, procedures, and methods of the process are applied." "The purpose of [quality] measurement and analysis (MA) is to develop and sustain a measurement capability used to support management information needs [1]."*

Keywords: *continuous improvement, measurement, metrics, process improvement, quality, standards*

Context

There are multiple perspectives on both quality and its measurement that depend on the stakeholder's point of view. Knowledge of these perspectives is important when recommending quality or measurement programs for a government organization.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to recommend how to establish a quality assurance program in the government systems acquisition or government operational organization. They are expected to be able to guide the establishment and direction of quality assurance programs, conduct process and product reviews, and influence the resolution of corrective actions to ensure adherence to documented processes. MITRE SEs are expected to be able to help develop measurement capabilities to monitor processes and products [2].

Perspectives on Quality

Some of the perspectives on quality are as follows [3]:

- **Judgmental:** When referred to as the transcendent definition of quality, it is both absolute and universally recognizable, a mark of uncompromising standards and high achievement. You can't really measure or assess it—you just know it when you see it. Lexus and Ritz-Carlton are examples.
- **Product-based:** In this view, quality is a function of a specific, measurable variable and differences in quality reflect differences in quantity of a product attribute, such as threads per square inch or pixels per square inch. Bed sheets and LCD high-definition televisions are examples.
- **User-based:** Quality is defined as fitness for intended use, or how well the product performs its intended function. If you want an off-road vehicle for camping, a Jeep might suit your needs. If you want a luxury vehicle with lots of features, a Cadillac might better suit your needs.
- **Value-based:** From this perspective, a quality product is as useful as a competing product and sold at a lower price, or it offers greater usefulness or satisfaction at a comparable price. If you had a choice between two "thumb drives" and one offered one gigabyte of storage for $39.95 and another offered two gigabytes of storage for $19.95, chances are you would choose the latter.
- **Manufacturing-based:** Quality is the desirable outcome of engineering and manufacturing practice, or conformance to specifications. For Coca-Cola, quality is "about manufacturing a product that people can depend on every time they reach for it."
- **Customer-drive:** The American National Standards Institute and the American Society for Quality (ASQ) define quality as "the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs [4]." A popular extension of this definition is "quality is meeting or exceeding customer expectations."

Quality Assurance Versus Quality Control

There is an important distinction between quality assurance (QA) and quality control (QC). ASQ defines QA as "the planned and systematic activities implemented in a quality system so that quality requirements for a product or service will be fulfilled." ASQ defines QC as "the observation techniques and activities used to fulfill requirements for quality [4]." Thus QA is a proactive, process-oriented activity whereas QC is a reactive, manufacturing-oriented activity.

The focus of QA is putting good processes in place so that the quality will be "built into" the product rather than trying to "inspect quality into" the finished product.

Quality Standards and Guidance

The International Organization for Standardization (ISO) 9000 family introduces the concept of quality management, processes, certification, and continual improvement. ISO 9000 is the internationally accepted standard for quality management. It looks at manufacturing and customer-based perspectives of quality. The ISO 9000:2000 family is built on eight quality management principles: (1) Customer Focus, (2) Leadership, (3) Involvement of People, (4) Process Approach, (5) System Approach to Management, (6) Continual Improvement, (7) Factual Approach to Decision Making, and (8) Mutually Beneficial Supplier Relationships [5].

The ISO 9001:2000 (the basis for ISO 9001 certification or registration) states, "This International Standard specifies requirements for a quality management system where an organization a) needs to demonstrate its ability to consistently provide a product that meets customer and applicable regulatory requirements, and b) aims to enhance customer satisfaction through the effective application of the system, including processes for continual improvement of the system and the assurance of conformity to customer and applicable regulatory requirements [6]." ISO 9001 registration is critical to securing and maintaining business for both private and public sector contractors. The government recognizes ISO 9001:2000 as a "higher level" quality requirement and may invoke it in the contract under the conditions stated in Federal Acquisition Regulation Part 46.202-4, Higher Level Contract Quality Requirements. In these situations, the government is more interested in the contractor's quality management system (and its certification) than government inspection of a product under development.

Government acquisition organizations rarely have an independent quality assurance organization that oversees the quality of the government's work products or processes. The Capability Maturity Model Integrated for Acquisition (CMMI-ACQ) includes a process area (Product and Process Quality Assurance) that provides a set of goals and specific practices for quality assurance in an acquisition organization [7]. Both government and contractor development organizations have a similar CMMI for Development (CMMI-DEV) process area [8].

The government occasionally introduces a quality or process improvement initiative that receives emphasis for a while and is then overcome by events or forgotten. Several of these past initiatives include the "Suggestion Program," "Zero Defects," "Quality Circles," and "Total Quality Management." Some of the most recent initiatives are Department of Defense (DoD) Six Sigma, DoD-wide Continuous Process Improvement/Lean Six Sigma, and Air Force Smart Operations for the 21st Century [9, 10, 11]. Most of these initiatives deal with process improvement by eliminating waste, streamlining the processes, or instituting a more efficient way to perform a required task that results in cost avoidance or cost savings.

Perspectives on Measurement

All three CMMI models—CMMI-ACQ [12], CMMI for Services (CMMI-SVC) [13], and CMMI-DEV [14]—include a process area for Measurement and Analysis. "The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability that is used to support management information needs [14]." There are eight specific practices recommended in the models: 1) Establish Measurement Objectives; 2) Specify Measures; 3) Specify Data Collection and Storage Procedures; 4) Specify Analysis Procedures; 5) Collect Measurement Data; 6) Analyze Measurement Data; 7) Store Data and Results; and 8) Communicate Results [14].

Four categories of measurement are common to many acquisition programs in which MITRE is involved:

1. The quantitative performance requirements of user requirements and system performance requirements are measured in the operational and development test programs. Suggestions on key performance parameters can be found in Enclosure B of the Chairman of the Joint Chiefs of Staff Manual on the Operation of the Joint Capabilities Integration and Development System [15].
2. Technical Performance Measurement (TPM) monitors the developer's progress in meeting critical performance requirements over the life of the program where there is a development risk. The concept is further explained on the Office of the Secretary of Defense (OSD) TPM website [16].
3. Earned Value Management (EVM) monitors a developer's cost and schedule performance in cost reimbursement development contracts. Additional information can be found in the article "Earned Value Management" and in EIA-748A and the OSD EVM website [17, 18].
4. Process Metrics are associated with development processes like software development. A good approach to identifying the type of measurement needed and the proven metrics that support that measurement can be found on the Practical Software and System Measurement website [19].

There is a fifth category that may be involved if MITRE is assisting in developing performance-based logistics criteria for operations and maintenance efforts. OSD recommends five performance parameters: 1) Operational Availability; 2) Operational Reliability; 3) Cost Per Unit Usage; 4) Logistics Footprint; and 5) Logistics Response Time. These are cited in an OSD Memo on the subject [20].

For additional information, refer to the articles “Acquisition Management Metrics” and “How to Develop a Measurement Capability.”

Articles Under This Topic

The article “Establishing a Quality Assurance Program in the Systems Acquisition or Government Operational Organization” provides guidance on processes to assure that the right product is being built (customer-driven quality), that the product being built will meet its specified requirements (product-based quality), and that the product is suitable for its intended use (user-based quality).

The article “How to Conduct Process and Product Reviews Across Boundaries” provides guidance on assisting government and contractor organizations in documenting quality processes and work product specifications, and reviewing those processes and products.

References and Resources

1. CMMI-DEV, Version 1.2.
2. MITRE Systems Engineering (SE) Competency Model, Version 1, September 1, 2007, The MITRE Institute, Section 3.7, Quality Assurance and Measurement, pp. 45-46.
3. Evans, J. R. and W. M. Lindsay, 2008, *Managing for Quality and Performance Excellence,* 7th Edition, Thomson, Southwestern.
4. The American Society for Quality (ASQ) website.
5. *ISO 9000:2000,* 2000. *Quality Management Systems, Fundamentals and Vocabulary,* Second Edition.
6. *ISO 9001:2000,* 2000, *Quality Management Systems,* Third Edition.
7. CMMI-ACQ, Version 1.2.
8. CMMI-DEV, Version 1.2.
9. Matchette, Daniel R., “Six Sigma for the DoD,” *Defense AT&L Magazine*, July-August 2006, Vol. 35, No. 4, p. 19–21.
10. DoD 5012.42, May 15, 2008, DoD-Wide Continuous Process Improvement (CPI)/Lean Six Sigma (LSS) Program.
11. The Secretary of the Air Force and Chief of Staff of the Air Force, February 7, 2006, Air Force Smart Operations for the 21st Century CONOPS and Implementation Plan, Version 4.
12. CMMI-ACQ, Version 1.2.
13. CMMI-SVC, Version 1.2.

14. CMMI-DEV, Version 1.2.
15. CJCSM 3170.01C, Operation of the Joint Capabilities Integration and Development System.
16. OSD Technical Performance Measurement website.
17. *Earned Value Management,* ANSI EIA-748A Standard (June 1998 ed.).
18. OSD Earned Value Management website.
19. Practical Software and System Measurement website.
20. OSD AT&L Memo, 16 Aug 2005, Performance Based Logistics: Purchasing Using Performance Based Criteria.

Additional References and Resources:

Metzger, L., May 2009, *Systems Engineering Quality at MITRE*, The MITRE Corporation.

The MITRE Corporation, August 21, 2009, “Quality,” *MITRE Project Leadership Handbook*.

Definition: *Quality Assurance (QA) is "a planned and systematic means for assuring management that the defined standards, practices, procedures, and methods of the process are applied [1]."*

Keywords: *continuous improvement, process improvement, quality, standards*

QUALITY ASSURANCE AND MEASUREMENT

Establishing a Quality Assurance Program in the Systems Acquisition or Government Operational Organization

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to recommend how to establish a QA program in the systems acquisition or the government operational organization. They are expected to propose plans to resource, implement, and manage a QA program to enable a positive, preventive approach to managing the systems acquisition. They are expected to participate in integrated teams to create directives and plans that establish QA standards, processes, procedures, and tools [2].

Background

MITRE assists the government in preparing contract requirements for the acquisition of large systems from major information technology contractors. With few exceptions, these contracts must comply with mandatory provisions in the Federal Acquisition Regulation (FAR).

The definition of quality for government contracts is stated in the FAR Part 46.101: "Contract quality requirements means the technical requirements in the contract relating to the quality of the product or service and those contract clauses prescribing inspection, and other quality controls incumbent on the contractor, to assure that the product or service conforms to the contractual requirements." Thus, the government contract interpretation of quality (control) in most contracts for major systems is the manufacturing-based perspective: conformance to specifications.

MITRE's Quality Guidelines and Standards

MITRE's QA efforts should focus on the use of appropriate processes to assure that the right product is being built (customer-driven quality), that the product being built will meet its specified requirements (product-based quality), and that the product is suitable for its intended use (user-based quality). This aligns with the view of systems engineering quality in the "Systems Engineering Quality at MITRE" white paper. This paper states, "1) Degree to which results of SE meet the higher level expectations for our FFRDCs [federally funded research and development centers]—resulting in usability and value for end recipients; 2) Degree to which results of SE [systems engineering] meet expectations of our immediate customers—service and performance [3]." MITRE's QA efforts are particularly appropriate for the design and development phases of the product, especially software and one or few-of-a-kind systems, rather than concentrating on quality control (QC) in the production phase. For additional perspectives on quality, see the SEG's Quality Assurance and Measurement topic.

Best Practices and Lessons Learned

Use project and portfolio reviews. The use of MITRE project reviews can provide project leaders with additional perspectives and assistance from elsewhere in MITRE, when necessary. This is a form of leveraging the corporation. Portfolio reviews help maintain a focus on a sponsor's most important problems and provide an opportunity for cross-portfolio synergy among the projects in the portfolio.

Establish watchlists. Watchlists in various forms (e.g., major issues, significant risks, external influences) provide a vehicle to keep project leaders focused on issues that are likely to have a critical impact on their programs. They also keep MITRE and senior government managers aware of project issues that may require escalation. If done at an enterprise level, watchlists can keep individual programs aware of enterprise issues where programs can make a difference.

Perform Engineering Risk Assessments (ERAs). ERAs are constructive engineering reviews that identify and resolve issues or risks that might preclude program success. In the Department of Defense, ERAs are performed on Acquisition Category II (ACAT II) and below programs. The ERAs focus on solution appropriateness, SE progress health, and SE process health. In doing so, the ERA considers all aspects of systems engineering in acquisition, including engineering to establish sound technical baselines that support program planning and program cost estimation, technical resource planning, engineering management methods and tools, engineering performance metrics, engineering basis of estimate and earned value management, system design appropriateness, system design for operational effectiveness (SDOE), and other areas. The ERA methodology provides a tailorable framework for conducting ERAs to assist program managers and appropriate decision makers in preparation for milestone decision and other reviews.

Perform independent assessments. Independent assessments can include Gold Teams, Blue Teams, Red Teams, Gray Beard Visits, or process assessments like the Standard Capability Maturity Model Integration Appraisal Method for Process Improvement (SCAMPI). All of these assessments involve the use of external subject matter experts to provide an objective opinion on the health of a program or organization and its processes. An independent assessment can be used at any point in the program life cycle to provide insight into the progress and risks. For example, the assessment may be used to provide an independent assessment of a preliminary design or an assessment of the product as it enters integration and test. Independent assessments are typically proactive and intended to provide an early look at potential problems that may be on the horizon in time to take action and avoid adverse impact to the program. One of the best opportunities for an independent assessment is during the management turnover of a program or organization. This provides the new manager a documented assessment of the organization and a set of recommended improvements. The new manager has the benefit of a documented assessment of the mistakes or improvements made by prior management. For more information, see the SEG's MITRE FFRDC Independent Assessments topic.

Conduct peer reviews of deliverables. Having an external peer review of formal deliverables ensures that the delivered product makes sense and represents a MITRE position rather than an individual's opinion on a product provided to our sponsor. This is particularly important on small projects that are located in a sponsor's facility. It keeps our staff objective in providing advice to our sponsors.

Perform after-action reviews. After participating in the development of a critical deliverable or briefing, or position for or with our sponsors, meet with the MITRE participants and discuss what was executed well and what could have been better. This allows us to continuously improve how we serve the customer and capture lessons learned for others engaged in similar activities for their sponsors.

Identify key requirements. All requirements are not created equal, although that may be the first response when you ask the "priority" question. Whether it is the key performance parameters in an operational requirements document, the critical performance parameters in a system performance specification, or the evaluation factors for award in a request for proposal, identify the most important measures that will impact the final decision.

Use Technical Performance Measurement (TPM) in conjunction with risky performance requirements only. If a given performance requirement is within the state-of-the-art technology, and there is little doubt that the developer will be able to meet the requirement, do not use TPM. Focus on the "risky" performance requirements where it is important to monitor progress in "burning down" the risk.

Identify program risks and problem areas, and then identify metrics that can help. Do not take a "boilerplate" approach when specifying metrics to monitor your program. Identify the significant programmatic and technical issues and risks on the program, then select metrics that provide insight into the handling or mitigation of the issues and risks.

Do not identify all metrics at contract award, specify them when you need them. As program issues and risks change as a function of time, the metrics to monitor the handling or mitigation of the issues or risks should change as well. In the front end of a program, developer ramp-up is something to monitor, but it disappears when the program is staffed. In the testing phase of a program, defect density is important, but not particularly important in the front end of a program.

Do not require measurement data, unless you have an analysis capability. Do not implement a set of metrics, unless you have staff with the capability to analyze the resulting data and make program decisions. Two examples of data that is frequently requested, but there is no program staffing qualified for analysis, are software metrics and Earned Value Management (EVM) data.

References and Resources

1. Software Engineering Institute, Carnegie Mellon, "CMMI-Development," Version 1.2, accessed February 11, 2010.
2. The MITRE Corporation, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model," Version 1, Section 3.7, p. 45.
3. Metzger, L., May 2009, "Systems Engineering Quality at MITRE."

Additional References and Resources

Acquisition Community Connection, August 16, 2005, Acting UDS/AT&L Policy Memo: Performance Based Logistics: Purchasing Using Performance Based Criteria.

Acquisition Community Connection, February 7, 2006, Air Force Smart Operations for the 21st Century CONOPS and Implementation Plan, Version 4.

CMMI Product Team, February 2009, "CMMI for Services, Ver. 1.2-CMMI-SVC, V1.2," Software Engineering Institute, accessed February 11, 2010.

Department of Defense Directive, May 15, 2008, "DoD-Wide Continuous Process Improvement (CPI)/Lean Six Sigma (LSS) Program," DoD 5010.42.

"Earned Value Management," June 1998, *ANSI EIA-748A Standard.*

Evans, J. R., and W. M. Lindsay, 2008, *Managing for Quality and Performance Excellence,* 7th Ed., Thomson South-western.

International Organization for Standardization, 2000, *ISO 9000:2000, Quality Management Systems, Fundamentals and Vocabulary,* Second Edition.

International Organization for Standardization, 2000, *ISO 9001:2000, Quality Management Systems*, 3rd Ed..

Joint Chiefs of Staff, May 1, 2007, Operation of the Joint Capabilities Integration and Development System, CJCSM 3170.01C.

OSD Earned Value Management, accessed February 15, 2010.

OSD Technical Performance Measurement, accessed February 15, 2010.

Practical Software and System Measurement, accessed February 15, 2010.

Software Engineering Institute, Carnegie Mellon, "CMMI for Acquisition, Version 1.2," accessed February 11, 2010.

The American Society for Quality (ASQ), accessed February 11, 2010.

The MITRE Corporation, August 2009, *MITRE Project Leadership Handbook,* "Quality."

Definition: *Process and Product Quality Assurance are activities that provide staff and management with objective insight into processes and associated work products [1]. A Quality Management Process assures that products, services, and implementations of life-cycle processes meet organization quality objectives and achieve customer satisfaction [2].*

Keywords: *noncompliance, objective evaluation, process, process description, process review, product review, quality, quality assurance, quality management, work products*

QUALITY ASSURANCE AND MEASUREMENT

How to Conduct Process and Product Reviews Across Boundaries

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) conduct process and product reviews across boundaries in the government systems acquisition and/or operational organizations. In this role, they assist both government and contractor organizations to document quality processes and work product specifications. To ensure adherence to documented processes and work product specifications, MITRE SEs review government and contractor quality processes and products and contractor quality assurance programs; prioritize quality process improvement opportunities and corrective actions; report to key decision makers the results of process and product reviews; and elevate high-priority corrective actions [3].

Background

When projects pay attention to the quality of their products and the processes that produce them, they have a better chance of succeeding. When there is a government-to-contractor relationship established through a contract, MITRE's role needs to sharpen because the focus of our analysis and guidance then has two sides involved. We need to think through such situations carefully to give customers our best advice. To do that, MITRE SEs should look to a well-recognized common process framework like ISO/IEC 15288 [2] and established process improvement models such as the Capability Maturity Model Integrated® (CMMI) [1] to ground our analysis and guidance and to get both sides on a firm process foundation.

Both the government and contractors need a common framework to improve communication and cooperation among the parties that create, use, and manage modern systems so they can work in an integrated, coherent fashion. ISO/IEC 15288 is an international standard that provides this framework and covers the life cycle of human-made systems. This life cycle spans a system from idea conception to system retirement. ISO/IEC 15288 provides the processes for acquiring and supplying systems and for assessing and improving the life-cycle processes.

ISO/IEC 15288 defines outcomes that should result from the successful implementation of a Quality Management Process:

- Organization quality management policies and procedures are defined.
- Organization quality objectives are defined.
- Accountability and authority for quality management are defined.
- The status of customer satisfaction is monitored.
- Appropriate action is taken when quality objectives are not achieved.

Furthermore, the standard defines certain activities and tasks an implementation is expected to follow:

- **Plan quality management,** which includes:
  - Establishing quality management policies, standards, and procedures.
  - Establishing organization quality management objectives based on business strategy for customer satisfaction.
  - Defining responsibilities and authority for implementation of quality management.
- **Assess quality management**, which consists of:
  - Assessing and reporting customer satisfaction.
  - Conducting periodic reviews of project quality plans.
  - Monitoring the status of quality improvements on products and services.
- **Perform quality management corrective action**, which consists of:
  - Planning corrective actions when quality management goals are not achieved.
  - Implementing corrective actions and communicating results through the organization.

Again, both government and contractor processes should each perform, in some fashion, these types of activities in their respective domains. The government has an additional responsibility: to have insight into and oversight of the contractor's quality management process to ensure it is in place, it is performing, and defects are identified and removed as a matter of daily practice. Otherwise, the government could be on the receiving end of poor-quality products that impact its commitments to its customers.

Both the government and contractors need to improve their processes. CMMI was developed by a group of experts from industry, government, and the Software Engineering Institute (SEI) at Carnegie Mellon University. It is a process improvement approach to software engineering and organizational development that provides organizations with the essential elements for effective process improvement to meet business goals. It can be used to guide process improvement across a project, a division, or an entire organization.

CMMI's Process and Product Quality Assurance Process Area supports the delivery of high-quality products and services by providing project staff and managers at all levels with appropriate visibility into, and feedback on, processes and associated work products throughout the life of the project. It establishes expectations that a project's (or an organization's) quality assurance process will objectively evaluate processes and work products so that when non-conformance issues are identified, tracked, and communicated, resolution is ensured. For every process, CMMI further establishes generic expectations, which include, for example, the establishment of policy, plans, and monitoring. Having a common process improvement reference model gets both sides to think about process improvement in the same way, establishes a common framework and language, and promotes cooperation at any troublesome touch points between the two.

For related information, see the other articles in this Quality Assurance and Measurement topic and those in the topic areas Continuous Process Improvement, Contractor Evaluation, and MITRE FFRDC Independent Assessments.

Best Practices and Lessons Learned

Yes…quality is important for government work, too. Strive for the establishment of an independent, properly positioned quality management function on the government side. It needs to be positioned at a high enough level to have senior leadership's attention and not be bullied by project managers and midlevel management. To expect quality assurance only on the contractor side is not good enough.

Use standards and previous efforts in establishing your quality capabilities. Don't start from scratch in developing an organization's quality management process. Use a standard like ISO/IEC 15288 for starters. Check with other government organizations and take a look at their quality office, and what policies, processes, and templates they use. Stand on the shoulders of previous efforts and apply them to your situation.

Set quality standards up front. Make sure there are organizational standards established for project work process and products. It's tough to check quality if you don't have a standard to check against. Look to IEEE or your government department or agency standards and tailor them for your organization. Check with the program's prime commercial contractor for examples, but remember the perspective from which those standards were built...from the supplier side, so they'll need to be adjusted.

Ensure quality expectations are built into the contracts/task orders. Build the expectation of quality into your contracts and/or task orders. If you don't state it, you're not likely to get it. Expect products from the contractor will be checked against the agreed-on standard, and defects will be identified and removed before the products are delivered. In the contract, make sure the government can periodically check defect records, process appraisal results, defect tracking databases, and process improvement plans to see if they're in place and actively being worked. Don't just expect quality to magically appear from the contractor. Many times it does not unless they know the sponsor cares enough to be checking.

Trust...but verify. Once the contract arrangement is up and running, the contractor's quality management function is operational, and trust is beginning to solidify, the government may want to consider only periodic reviews of contractor processes on a sampling basis. Objective evidence of active reviews and defect resolution is key.

Have a common strategy for improving the process. On both the government and contractor sides, there should be active process improvement efforts in place that continuously look to mature the efficiency, effectiveness, and timeliness of their quality assurance activities. It is highly desirable that both sides agree on the same process improvement model, such as CMMI. Both sides should have a common vocabulary and improvement and appraisal methods, which promote effective communication and collaboration opportunities to help speed performance improvement initiatives.

Remember the bottom line. MITRE SEs should actively encourage and promote quality management processes and standards on both sides, properly positioned in their management structures, with a culture that encourages process improvement to ultimately result in higher quality, on-time, and useful systems.

References and Resources

1. CMMI Product Team, CMMI for Development, Ver. 1.2, CMU/SEI-2006-TR-008.
2. ISO/IEC 15288, IEEE Systems and Software Engineering—System Life Cycle Processes.
3. The MITRE Corporation, "MITRE Systems Engineering (SE) Competency Model, Version 1," September 1, 2007, p. 46.

Continuous Process Improvement

Definition: *A process is a set of steps to accomplish a defined purpose or produce a defined product or service. Continuous process improvement is the set of ongoing systems engineering and management activities used to select, tailor, implement, and assess the processes used to achieve an organization's business goals. Continuous improvement is recognized as a component of modern quality management [1].*

Keywords: *continuous process improvement, plan-do-check-act cycle, process-based management, process improvement, process model, systems engineering processes*

Context

The state of the art in system development management has evolved over the last few decades from basic concepts, practices, techniques, and tools borrowed from other disciplines to a relatively sophisticated suite of training, guided experience, and performance evaluation using structured collections of proven best practices. Experience has shown repeatedly that careful planning, frequent, regular review by trained, qualified people, and meticulous control of product components as they are developed, while not automatically sufficient by themselves, are necessary to defining and fielding a complex product or system today. The technology product and service industry as a whole has attempted numerous times to define, document, and disseminate collections of sound practice and specifications of product quality. These have taken the form of standards, specifications, methods, tools, books, and training and certification programs, among others.

MITRE SE Roles and Expectations

MITRE systems engineers (SEs) are expected to be able to collaborate with sponsors and clients to develop and influence the government's approach to implementing and improving systems engineering processes for the supported acquisition organization. They are expected to be able to draft policy, develop plans, and conduct maturity assessments for the technical and engineering processes. MITRE systems engineers are expected to be able to collaborate with government and contractor organizations to implement, assess, and improve shared systems engineering processes [1].

A Four-Step Process

Despite the ever changing, ever more sophisticated forms of delivery and media, success in managing the development and operation of complex technology-based systems is still based on a well-executed "plan-do-check-act" cycle. It is founded on the quality control research of mathematician Dr. Walter A. Shewhart conducted in the United States during the 1940s, 50s, and 60s and broadened and elaborated by many others including, most notably, W. Edwards Deming [2, 3, 4, 5].

Simply stated, the cycle is a four-step process used to control product quality during the development process. The steps are to: (1) Plan: determine what needs to be done, when, how, and by whom; (2) Do: carry out the plan, on a small scale first; (3) Check: analyze the results of carrying out the plan; and (4) Act: take appropriate steps to close the gap between planned and actual results. Then, repeat, starting at Step 1.

"What needs to be done" is often expressed in the form of a process. Systems engineers (SEs) translate the concept of "small-scale first" into producing a prototype, a model, simulation, or mockup, or conducting a pilot project or trial run before producing the full-scale version or initiating production. They build in regular review, measurement, and evaluation of the resulting work products and the plans and processes used to build them. Then they act to take corrective action as deviations from plans and expected results emerge—or as potential deviation is predicted based on quantitative analysis of actual results against the background of prior experience.

Process-based Management

This is process-based management. Using a systems engineering process-based approach, planners, project managers, engineers, and other technical staff decompose the work of defining and building large, complex systems into more manageable, repeated cycles of these four steps. Innovators and researchers are still looking for and proposing better approaches but, for now, this is one of best we have found.

Processes may be thought of as generic templates for the components of specific plans. They document the best way an organization knows how to do something. Mature organizations manage and control them as they do other valuable tangible assets. Properly structured, documented processes clearly identify the work product or service to be produced or provided, along with the inputs required, measurements that will be applied to determine compliance and quality, and any specific methods, tools, and training available. Entry and exit criteria indicate the conditions that prompt initiation of the process and those that help to determine when it is finished.

Systems engineers select and sequence individual process descriptions to implement system development life-cycle models and corresponding work breakdown structures and to organize and tailor a technical approach to a particular project's needs and circumstances. If documented processes have been used, measured, and refined repeatedly—that is, if they have been continuously improved—systems engineers and cost estimators should be able to ascertain with some degree of confidence how long it will take to perform the processes again with a given set of resources, requirements, and other constraints.

Articles in This Topic

The article "Implementing and Improving Systems Engineering Processes for the Acquisition Organization" provides guidance on commonly used systems engineering processes available to assist MITRE SEs in developing organizational process policies and plans and conducting maturity assessments. The article emphasizes that effective systems engineering efforts require both the government and contractor organizations to continuously mature and improve processes.

The article "Matching Systems Engineering Process Improvement Frameworks/Solutions with Customer Needs" provides guidance on working with the government to select and tailor a process model appropriate to the task at hand. The article highlights two important process improvement issues: working on the systems engineering/technology problem, and developing and executing a strategy to orchestrate associated organizational change.

Continuous process improvement is closely associated with quality assurance and viewed by many as an aspect of it. For related information, see the SEG's Quality Assurance and Measurement topic.

References and Resources

1. The MITRE Institute, September 1, 2007 “MITRE Systems Engineering (SE) Competency Model, Version 1,” pp. 47–48.
2. The Project Management Institute, 2008, *A Guide to the Project Management Body of Knowledge,* (PMBOK Guide), 4th Ed., pp. 189–191.
3. Kerzner, H., 2003, *Project Management,* 8th Ed., New York: John Wiley & Sons, Inc., pp. 761–765.
4. Shewhart, W. A., 1931, *Economic Control of Quality of Manufactured Product,* New York: D. Van Nostrand Company.
5. Deming, W. E., August 2000, *Out of the Crisis,* Cambridge, MA: MIT Press.

CONTINUOUS PROCESS IMPROVEMENT

Implementing and Improving Systems Engineering Processes for the Acquisition Organization

Definition: *Project management and systems engineering should be integrated into a seamless set of processes, plans, work products, reviews, and control events that are documented and continuously improved in an orderly, controlled manner, based on experience and lessons learned.*

Keywords: *Capability Maturity Model Integration (CMMI), continuous process improvement, process, process improvement, process model, Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal, systems engineering processes*

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to support the implementation and continuous improvement of core and shared systems engineering processes. They develop process implementation plans, including process improvement goals, schedules, and estimated resources, and they identify the need for, and often assist in, conducting process maturity assessments [1].

The MITRE SE's role can vary from that of a trusted advisor providing guidance to the government on critical government and contractor work products, processes, methods, and tools to direct involvement in the development of strategies, plans, technical specifications, statements of work, methods, tools, and commercial off-the-shelf product evaluations and recommendations, coaching, training, and decision support. The MITRE SE should assume a degree of ownership for the effectiveness of the systems development process. The MITRE SE should assist the government in achieving organizational performance goals, provide constructive feedback to development contractors in collaboration with the government, and help assure the quality, integrity, and appropriateness of MITRE's products and services.

Background

A process is a set of steps to accomplish a defined purpose or produce a defined product or service. The state-of-the-art technical aspects of systems development and management have evolved over the past few decades from basic concepts, practices, techniques, and tools borrowed from other domains into a sophisticated, structured engineering discipline called "systems engineering." Experience shows that supporting disciplined program and project management with rigorously applied systems engineering is a steadfast approach to successfully defining and managing the development and fielding of complex technology-based products and services. The most effective way to implement this strategy is by integrating project management and systems engineering into a seamless set of processes, plans, work products, reviews, and control events that are documented and continuously improved in an orderly, controlled manner, based on experience and lessons learned.

Government agencies typically obtain software-intensive systems, hardware, facilities, and operational support by issuing contracts for services from commercial contractors. The government calls this approach "systems acquisition." The government's role is to define what it wants to acquire, how it will acquire it, and how to plan and manage the effort on behalf of its end-user organizations and operators. If commercial bidders hope to be awarded a contract, they must demonstrate prior experience and success with the products and services sought by the government as well as exhibit the required management, technical, and support services skills.

The government also has realized that it must have the capability to perform its role effectively and work in partnership with the selected contractor; the government sets the tone for the partnership. What the acquisition organization does is just as important as what the system development contractor does. An immature, demanding, dysfunctional acquisition management organization can render ineffective the practices and potential of a mature, high-performing contractor. When one or both parties perform inadequately, the entire development process is impaired. Planning and documenting what each party will do, how each will do it, and when each will do it is essential to success. Each party needs to keep activities and work products up to date and synchronized. Plans and methods need to be refined as more is learned about the nature and challenges inherent in the system or capability being built and its intended operating environment.

Systems Engineering Process Improvement

Systems engineering processes are documented in a variety of sources, including the International Council on Systems Engineering TP-2003-002-03.1 Systems Engineering Handbook, Institute of Electrical and Electronics Engineers (IEEE) Std. 15288-2008 Systems and Software Engineering—System Life Cycle Processes, and American National Standards Institute (ANSI)/Energy Information Administration (EIA)-632-1999 Processes for Engineering a System.

Most of these references cite and implement project management, technical control, and systems engineering guidelines collected in the Carnegie Mellon Software Engineering Institute's CMMI for Development, Version 1.2, 2007. The CMMI is organized by various categories: Process Areas, Maturity Levels, and formats. See the Process Areas in the "Engineering" category, primarily in Maturity Level 3 of the "staged" version of the model.

Carnegie Mellon University's IDEAL model (Initiating, Diagnosing, Establishing, Acting, and Learning) is a widely used method for project management and systems engineering process improvement. This model serves as a roadmap for initiating, planning, and implementing improvement initiatives or projects. Process improvement professionals can apply it with the CMMI and the SCAMPI. Note that the IDEAL model is a variation on Deming's "Plan, Do, Check, Act" cycle, which is discussed in the article "Continuous Process Improvement" [2].

Depending on the need, several other prominent process improvement methods are available. Examples include the International Standards Organization (ISO) 9000 Series, the Information Technology Infrastructure Library, and the ISO/International Electro-technical Commission 15504 standard, also known as the Software Process Improvement and Capability Determination method.

Some government agencies have adopted "best practice" process models (e.g., Carnegie Mellon Software Engineering Institute's CMMI or various ANSI and IEEE standards) as guidelines for assessing a contractor's system development capability and performance. Some agencies have gone further. They acknowledge that by adopting versions of these recognized frameworks and guidelines tailored to their respective roles and responsibilities, they can contribute more to reducing the risks that are inherent in complex systems development and deployment. In this type of operating environment, the MITRE SE can assess system design and development plans, processes, and actual activities against the requirements of the process model being used. The MITRE SE also can determine the level of compliance, identify gaps, and recommend remedial actions to the government and, indirectly, to development contractors.

Best Practices and Lessons Learned

Consider adopting a de facto measurement standard or benchmarking tool when your organization does not use a recognized process model. This best practice requires judgment. If you tailor a de facto model, recommend incremental changes or additions to current practice that are feasible without substantial impact on schedules and resources. Focus on recommendations that reduce specific acknowledged risks or contribute to resolving or preventing the recurrence of specific known issues. The process model should be used as a checklist and part of your knowledge base, rather than as a binding standard.

Base recommendations for process improvement on a recognized process improvement framework. A structured improvement is most effective when based on a recognized process framework and on proven organizational change management or organizational development methods guided by trained, experienced organizational change and process improvement professionals. See the article "Matching Systems Engineering Process Improvement Frameworks/ Solutions with Customer Needs" for guidance on selecting an appropriate process model. Get help with the intricacies of organizational change management and process improvement if you or your team have not already demonstrated mastery of these methods and tools.

References and Resources

1. The MITRE Corporation, September 1, 2007, "MITRE Systems Engineering (SE) Competency Model, Version 1," pp. 48-49, accessed February 10, 2010.
2. Deming, W. E., August 2000, *Out of the Crisis*, MIT Press, Cambridge, MA.

Additional References and Resources

Chrissis, M. B, M. Konrad, and S. Schrum, 2007, *CMMI, Second Edition, Guidelines for Process Integration and Product Improvement*, The SEI Series in Software Engineering, Boston, MA: Pearson Education, Inc., Addison- Wesley.

Chrissis, M. B., M. Konrad, and S. Schrum, 2003, *CMMI: Guidelines for Process Integration and Product Improvement*, Boston, MA: Pearson Education, Inc., Addison-Wesley.

McFeeley, R., February 1996, *IDEAL: A Users Guide for Software Process Improvement Handbook,* CMU/SEI-96-HB-001, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Shewhart, W. A., 1931, *Economic Control of Quality of Manufactured Product*, New York: D. Van Nostrand Company.

Definition: *Frameworks that enable systems engineering process improvement provide a basic conceptual structure to solve or address complex issues by designing, establishing, refining, and forcing adherence to a consistent design approach [1].*

Keywords: *business performance model, improving efficiency, organizational maturity, process-driven management, process improvement, quality management, systems engineering best practice*

CONTINUOUS PROCESS IMPROVEMENT

Matching Systems Engineering Process Improvement Frameworks/ Solutions with Customer Needs

MITRE SE Roles and Expectations: MITRE systems engineers (SEs) are expected to collaborate with government and contractor organizations to select and tailor systems engineering process improvement models—e.g., Software Process Improvement and Capability Determination (SPICE), Software Engineering Institute (SEI) Ideal, Capability Maturity Model Integrated (CMMI), Lean Six Sigma, etc. These are used to modify, integrate, test, baseline, deploy, and maintain systems engineering processes for the government acquisition and/or contractor organizations [2]. SEs should be aware of the spectrum of choices for continuous process improvement (CPI) efforts, be able to form recommendations about them, and assist in implementing a selected approach within their work environment.

Background

Each process improvement framework brings its own set of standards and strengths to satisfy customer needs. Some, such as CMMI, Control Objectives for Information and related Technology (COBIT), and Information Technology Infrastructure Library (ITIL) come from a set of best practices. Others, such as Lean Six Sigma, consist of strategies or tools to identify weaknesses and potential solutions. Because systems engineering process improvement frameworks overlap, more than one framework may match the customer needs for a CPI effort. These frameworks are available for any effort; there are no exclusivity rights. For example, the ITIL characterizes itself as "good practice" in IT service management. The SEI has a CMMI for services [3].

There are standard sets of processes that provide templates and examples of key processes such as IEEE's ISO/IEC 15288 [4]. The Six Sigma, Lean Manufacturing, Lean Six Sigma family of frameworks each contain tools to assess and improve processes, and are currently in use in government organizations.

CMMI has gained a great deal of popularity over the past few years. The Government Accountability Office (GAO) has been basing its oversight reviews on this framework and the results are flowing back to the departments and agencies with references to CMMI best practices. As a result, some in government are taking the view that an organization aligned with CMMI best practices and certified for its software development processes' level of maturity at 2 or 3 will receive greater approval from the GAO and other oversight groups. This has promoted CMMI's influence.

Lean Six Sigma is growing in popularity. It represents the joining of Six Sigma and Lean Manufacturing. Six Sigma was heavily touted a few years ago in government and industry and is still used in some sectors because of the methodology's success in eliminating defects. However, the downside was that it took too long and was too expensive. Lean Six Sigma, as the name implies, is faster to complete and requires fewer resources. The combination of Six Sigma and Lean tools and techniques is more effective and efficient and contains a richer solution set.

Selecting a framework may be based on factors that do not relate to the problem being addressed. Popularity of the framework can come into play. The background and experience of the individual leading the CPI effort can influence the approach. The customer may have some predetermined ideas as well.

Matching a process improvement framework/solution to the customer needs involves two issues: working on the systems engineering/technology problem, and developing and executing a strategy to orchestrate any associated organizational change. Any solution will require some members of the organization to perform their duties differently. Continuous process improvement often has a long-term, ongoing impact, as the processes are refined. Individuals in an organization executing CPI need to be comfortable with the change and embrace it.

Some of the frameworks are concerned with "what" should be done and others focus on "how" it should be done. Frameworks such as CMMI are in the "what" category. For example, CMMI indicates the need for a requirements development process area. It involves establishing customer and product requirements and analysis and validation. However, it does not prescribe elements such as what approach to use, the process model, or what rules to follow. Frameworks such as Six Sigma are in the "how" category. Six Sigma suggests appropriate tools to arrive at the right solution. Examples for requirements development include house of quality, voice of the customer, and affinity diagrams tools.

There is a high percentage of failure or slow progress in CPI efforts. In a 2007 quarterly report, SEI reported that it takes on average 20 months to attain CMMI Level 2 and 19 months to attain CMMI Level 3. The variation of time to attain Level 2 and Level 3 is large. [5] Many organizations fail on their first attempt and have to restart before they are successful. This is a consistent pattern with CMMI implementation. If there is frequent change in leadership at the top of the organization, the risk for completion is higher because new leadership often brings new direction.

Best Practices and Lessons Learned

Consider the path of least resistance. Be a pragmatist but do not give up on principles. There are many ways to meet customer needs. If you have built a trusting relationship, you can guide the way customer needs are met through the appropriate approach. When more than one framework will do the job, do not get hung up on which framework is "best" (e.g., ITIL [6] versus CMMI for Services [7]). If there is a positive history of ITIL in the organization and it fills the need compared to a CMMI for services solution, evaluate whether the benefits you might gain outweigh the costs and difficulty of making the shift. When you are assessing alternative approaches, when the more difficult path is the only way to accomplish the client goals completely, then advise the client accordingly and include a clear and frank discussion of the difficulties.

Critically consider the customer's true need. Beware of falling into the trap of investing time defining a complete program-level set of policies, charters, data repositories, metrics, processes, procedures, etc., if the customer really only needs processes specific to each project and has no desire to be certified at any level.

Organizational change is the difficult step. Implementing CPI normally involves a very different way of doing business in a continuously changing environment. CPI may not be universally viewed as an improvement by those affected by the change. Consider bringing in an organizational change expert.

Combine approaches. There may be strength in a combined or hybrid CPI approach. As an example, the CMMI framework is built on best practices from a variety of government and industry sources that do a very good job of explaining "what" to do, but do not provide guidance on "how" to do it. For example, for a certain level of maturity, CMMI requires a requirements development process. However, it does not define how to do that process. If you are looking for predefined processes, consider ISO 9000. [8] If you are looking to create your own, consider Lean Six Sigma and tools like voice of the customer, affinity diagrams, or house of quality.

Gain and use upper management support. Elicit and gain upper management support to settle on the right framework/solution for the organization before attempting to implement it. This is crucial regardless of which framework is selected. Use a top-down strategy to promote the CPI program. A bottom-up approach alone rarely results in a successful outcome. Even if it is successful, the project will usually take much longer.

Avoid labeling processes as new. Embed the process improvement effort into the normal way the organization conducts business. Avoid calling attention to it by using the framework name. Make the process part of refining the organization's customary system development life cycle. Otherwise, you risk creating the assumption that workloads will be enlarged. Compliance is more likely when those affected understand the change is merely a refinement to the work they are already doing.

Conclusion

There are many reasons why an organization may not gain traction when adopting a CPI program. When an implementation falters, do not automatically assume it is due to the chosen framework or approach. Ask yourself whether there really is a compelling reason for CPI or if there is a lack of engagement or buy-in by the customer. If so, it may be better to defer until another time.

References and Resources

1. Statemaster.com Encyclopedia, accessed February 12, 2010.
2. The MITRE Corporation, "MITRE Systems Engineering (SE) Competency Model, Section 3.8."
3. CMMI Product Team, August 2006, CMMI® for Development, Version 1.2, Carnegie Mellon Software Engineering Institute.
4. ISO/IEC 15288 IEEE, February 1, 2008, Systems and software engineering—System life cycle processes, 2nd ed..
5. Software Engineering Institute (SEI), 2006, SEI Annual Report, Carnegie Mellon.

6. Bajada, S., February 2010, *ITIL v3 Foundations Certification Training*. This book and foundations courses for the Information Technology Instruction Library (ITIL) are available through The MITRE Institute's SkillPort e-Learning site.
7. CMMI Product Team, February 2009, CMMI® for Services: Improving processes for better services, Version 1.2, Carnegie Mellon Software Engineering Institute.
8. International Standards Organization (ISO), January 2009, ISO 9000.

Index

The MITRE Corporation's Systems Engineering Guide conveys MITRE's accumulated expertise on a wide range of systems engineering subjects—ideal for understanding the essentials of the discipline and for translating this guidance into practice in your own work environment. The online version of MITRE Systems Engineering Guide (on www.mitre.org) has been viewed by hundreds of thousands of people around the world. Here is what readers are saying:

The MITRE Corporation is a not-for-profit organization that operates federally funded research and development centers (FFRDCs). FFRDCs are unique organizations that assist the U.S. government with scientific research and analysis, development and acquisition, and systems engineering and integration. We're proud to have served the public interest for more than 50 years.

"This guide should help the entire systems engineering community significantly."

"This is simply excellent—it is a fantastic step forward to empowering me and others like me 'in the field.' Well done!"

"The Systems Engineering Guide fills an important niche for systems engineering practitioners."

"It is obvious that MITRE has put a significant amount of effort into the guide, and it is a valuable contribution to the systems engineering community."

"I will use the Systems Engineering Guide as a resource in teaching and research."

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
(781) 271-2000
7515 Colshire Drive
McLean, VA 22102-7539
(703) 983-6000
segteam@mitre.org
www.mitre.org

Covering more than 100 subjects, the guide's articles are written by MITRE systems engineering practitioners with substantial experience in particular subject areas. Each article identifies real-world problems that commonly occur in engineering systems and provides best practices for avoiding and mitigating them.

$0.00
ISBN 978-0-615-97442-2
90000>
9 780615 974422

MITRE