The Architecture of Enterprise Architecture

Terence Blevins
MITRE Corporation
7515 Colshire Drive
McLean, VA 22102-7508

Abstract. The Architecture of Enterprise Architecture section, answers questions such as; what makes up enterprise architecture? where is enterprise architecture relative to other enterprise processes? and what is necessary to make enterprise architecture work in the complex environments where it is deployed? This paper presents a categorization of architecture offerings that can be used to position and describe the architecture space and provides summarized and supportive information to the main thesis including architecture in the value chain, the social framework in which architecture exists, the skills required for success, and a view of the roles and responsibilities of architects.

Acknowledgements. I would like to acknowledge Murray E Daniels, Eugene Pierce, and Diane Hanf who each have contributed valuable content and comment to this paper.

INTRODUCTION

This Architecture of Enterprise Architecture paper answers questions such as:

- What makes up enterprise architecture?
- Where is enterprise architecture relative to other enterprise processes? and
- What is necessary to make it work in the complex environments where it is deployed?

The first section "Carving up the Architecture Space." presents a categorization of architecture offerings that can be used to position and describe the architecture space. This section represents the bulk of this paper; the other sections are summarized and supportive information. The additional sections are described below.

Realizing that architecture is a means to an end, not an end unto itself, a description of architecture in context to the value chain is included in the section "Where is Architecture in the Value Chain?"

The section "Social Framework or Perspective of Architecture" elaborates on the environment typical of the need for architecture. It provides an appreciation of the need to understand the different social aspects that are critical to obtaining buy-in to the architecture. This includes understanding the customers and the developers, and being able to provide a communications bridge between them.

The section on "Architecture Leadership" provides additional insights on the characteristics of leaders of architecture. Those characteristics apparent in those architects that have successfully used architecture are described here.

To complement the subject of Architecture Leadership there is a section on required architecture skills in "Architecture Skills." This section provides architects with a list of skills, that when honed, best prepare an architect for the task at hand.

The “Architect Roles and Responsibilities” section summarizes the accountabilities that an architect should expect. This provides the budding architect with an understanding of what is typically expected.

TERMINOLOGY

The following are definitions or descriptions of key terminology used in this document.

There are many definitions of architecture; the definition stated below is fit for use in this document.

> **Architecture –**The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. (CJCSI 3170.01D – 12 March 2004)

A definition for architecture view is also needed; the following definition is from the IEEE accepted definition of view.

> **Architecture View –** A representation of a whole system from the perspective of a related set of concerns. (IEEE STD 1471-2000)

Much like stated in IEEE STD 1471, architecture is described through a set of views in an architecture description.

Finally germane is the notion of “integrated architecture” where the following JCIDS definition is used.

> **Integrated Architecture** – An architecture consisting of multiple views or perspectives (operational view, systems view and technical standards view) that facilitates integration, promotes interoperability, and permits identification and prioritization of capability shortfalls and redundancies. (CJCSI 3170.01D)

Armed with these definitions the following formulates some views of the architecture of enterprise architecture in the first section “Carving up the Architecture Space.”

CARVING UP THE ARCHITECTURE SPACE

The industry is full of work that either contains the word architecture or is referenced as architecture. There are offerings called Enterprise Architecture Planning, and various Enterprise Architecture Frameworks, and Architecture Models. Each of these may be well understood within the community, or sub-community, of architects, but the use of the term “architecture” in all these different contexts confuses those not deeply involved.

This section presents categories of architecture offerings that can be used to position and describe the architecture space.

The perspective being taken is that of a person or organization with a need for architectural help, the client, which is being confronted with a person or organization that has an offering, the provider. The problem that the client has is understanding the provider’s offering so they can compare apple to apples when looking at multiple offerings. The following presents a way to look at the offerings within a set of unambiguous categories representing a proposed architecture of architecture.

These categories are intended to answer the following questions:

- Is the provider offering justification, or is it answering **why** architecture is necessary?

- **What** real architecture parts is the provider offering?
- Or is the provider is offering people or **who** delivers an architecture service?
- Or is the provider offering a method, a **how** one does architecture?

As a point of caution, it is very easy to lose focus and debate continuously about what goes in what category. For example everything one does could be considered “a what” from the perspective of “what I’m doing.” This must be avoided and instead we need to embrace the spirit of the categorization to benefit from it. In other words think about the tangible goods that are being delivered, the people that are delivering, the methods being used, and the justifications from the clients’ perspective.

Each of these categories has further subcategories that can provide even greater distinctions and are described in the following sub-sections.

The Whys – The Purpose

From the clients’ perspective a key problem area is understanding how to justify an architecture project. A provider can assist by delivering services or providing case studies to develop and articulate the rationale for an architecture project. These offerings fall in the purpose category.

The purpose category captures saleable justifications for doing architecture. Included here is market research data, case studies, cost benefit analysis, etc. This category is augmented by a description of the benefit when architecture is done in pre-development, post-development, and/or during ongoing governance. The sub-categories are described below.

Architecture **case studies** provide saleable descriptions of overall architecture engagements. These case studies represent valuable knowledge that can be transferred without an architecture consultant or architecture practitioner and allow an organization to determine whether it needs an architecture, what an effort entails, and whether it can do architecture on its own.

Architecture **cost benefit analysis** provides a means for organizations to determine the benefits of doing architecture in their organization. This helps the organization justify how architecture would improve the business.

Architecture **research data** provides basic statistics about doing projects with an architecture driven approach as opposed to not doing a project with an architecture approach. This data objectively supports decisions to move forward with architecture projects.

The Whats – Architecture Content

Another challenge a client has is understanding the deliverables of an architecture project. A provider can easily overwhelm a client with the different ways of describing the forms that architecture content will take. The following is an attempt to de-mystify this.

The architecture content category is complex since there are different levels of “things” that can be offered in this category. For example actual architectures, whole or part, can be offered. Alternatively products called reference models or frameworks can be offered. To organize this, the following describes the subcategories of this area.

First there are actual re-usable **architecture work products**, e.g. complete or partial architecture descriptions or other artifacts. These would include fully described architectures or architecture building blocks. These are architecture products that are directly re-usable in the creation of an architecture description. They are "parts" of architecture and would include actual models. Other products in the subcategory include re-usable architecture principles or architecture heuristics, and architecture patterns. Within the Department of Defense DODAF artifacts would be in this sub category.

Second there are more abstract descriptions that can be re-used – **architecture viewpoints**. These include descriptions of stakeholders, their concerns, and the model descriptions one would use to deal with these stakeholders and their concerns. [1471 is a standard for proper description of a viewpoint.] [MDA PSMs and PIMs are examples of viewpoints associated with software development.]

Third there are **architecture frameworks** that package sets of viewpoints. Some frameworks are more "complete" than others. [Zachman Framework is one of these, The Open Group's taxonomy of viewpoints is another example.] Within the US Federal Government Federal Enterprise Architecture there are specific frameworks each identified as reference models[1] including Performance, Business, Service Component, Technical, and Data Reference Models.

Taking these concepts together one can envision a hierarchy:

- Architecture Frameworks containing
    - One or many Architecture Viewpoints depicted through
        - One or many Architecture Work Products

The Whos – The Architects

Clients are often barraged with different types of architects. The following describes the architect and the services they provide.

The who category contains the architect (their roles and implied skills they have) and includes architecture consultants and/or architecture practitioners that deliver architecture services. Architecture services are also considered a sub-category of this category.

There are literally dozens of different types of architects that can be presented to a client as anything can be architected. To deal with this the following provides a number of dimensions that can be used to bucket architects (and their services) to help make reasonable comparisons.

The first dimension is that of **scope**; the concepts of enterprise, system of system, system, or component can be used to understand different scopes without a great deal of definitions.

The second dimension is that of **specialization**; doctrine (including security), organization (including process), training, material (including information, networks, hardware, software), leadership, personnel (including roles and skills), and facilities are high level dimensions adopted by DoD.

The third dimension is the nature of the architecture product **need**ed; architecture consultants provide advice where architecture practitioners produce architecture product. Architecture consultants are those people that provide services that result in advice, such as accept this architecture, rework this architecture, focus the architecture here and not there. The architecture consultant is first and foremost

[1] The term reference model is really poorly chosen. A reference model is typically considered an implementation of a model that can be referenced as a good or bad implementation, a baseline. The context of the usage of the term in FEA leads us to map the concept to framework.

an analyst. Architecture practitioners are those people that deliver architecture services that result in architecture products. The architecture practitioner is first and foremost an architecture developer.

Architecture consultants and practitioners can be classified, and even certified, as enterprise architects, system architects, or architects of a special area such as networking, database, security, performance, etc.

The above three dimensions can be represented in the following table.

	Doctrine		Organization		Training		Material		Leadership		Personnel		Facilities	
Enterprise														
System of System														
System														
Component														
	Con	Pra	Con	Pra	Con	Pra	Con	Pra	Con	Pra	Con	Pra	Con	Pra

Con – Consultant

Pra – Practitioner

Table 1: The Categories of Architects

The Hows [2]– The Processes, Methods, and Tools

From a clients' perspective a key problem area is understanding the various processes, methods, and tools that a provider presents them. The client is often presented with various methods purported to be better than another. It is very difficult to assess these different methods; the following provides a representation of some of the methods and help position each.

This category has many elements and informs a client on the various ways that one does architecture, for example through methods, processes, procedures, plans, and tools. This category includes anything that helps one "architect." In many cases a particular development method has a customized process, and/or set of procedures, plans, and tools that support the optimal execution of the method. The following are the primary sub-categories.

Architecture method and processes are the highest-level sub-category described, again because many of the other sub-categories are specialized to support a given method and process. Architecture methods are descriptions of an approach to follow when architecting. There are many methods, for example principles based architecture, business based architecture, technology based architecture, etc. Methods assume a specific approach such as being inspired by a business, being driven by new technologies, being driven by information technology standards, being driven by information technology principles, etc.

Executing an architecture engagement is supported by **architecture project plans**. Architecture plans should not be confused with architecture artifacts, the plans discussed here are the management plans one uses to manage an architecture engagement. Again these plans are typically aligned with a

[2] In this section we do not describe the processes; this is covered in other sections. Here we simply provide some types of the processes, methods and tools that exist.

specific architecture method and process, and deliverables set, although one can not discount the possibility of a generic plan.

The final sub-category is **architecture tools** which contains computing software that support the creation, storage and retrieval of architecture artifacts. This not only includes a typical architecture tool, but also anything that supports the creation, storage, and retrieval of architecture artifacts, including things such as schemas, languages, and repositories that facilitate the storage and re-use of architecture artifacts. [IEEE 1471 might be considered as being part of this.]

There are numerous methods, tools, and technologies used today.

- Architecture modeling methods and modeling tools
- Risk modeling methods and modeling tools
- Collaborative engineering environment
- Software modeling methods and modeling tools
- System engineering methods and engineering tools
- Cost modeling methods and modeling tools
- Process modeling methods and modeling tools
- Information integration methods and integration tools
- Simulation methods and tools

All of these methods and tools *are*, or better put, *can* be supported by:

- Architecture ontologies
- Architecture repositories
- Architecture federation technologies
- Data management tools, models and sharing tools

Today the situation is that these methods, tools, and technologies are all germane to the architecture and system engineering disciplines necessary to deliver capabilities by dealing with the subjects of **capabilities, costs, behaviors, requirements, and dependencies**.

What remains to be done in this area is to map these processes, methods and tools to such enterprise processes such as requirements management, portfolio management, program planning, and acquisition processes.

Completed Categorization Scheme

The following table summarizes the category scheme that covers the different architecture offerings, in part or whole. This can help a client understand what is actually being proposed by a provider, and based on need can help the client **decide** on what they actually need and then whether an offer is fit for purpose.

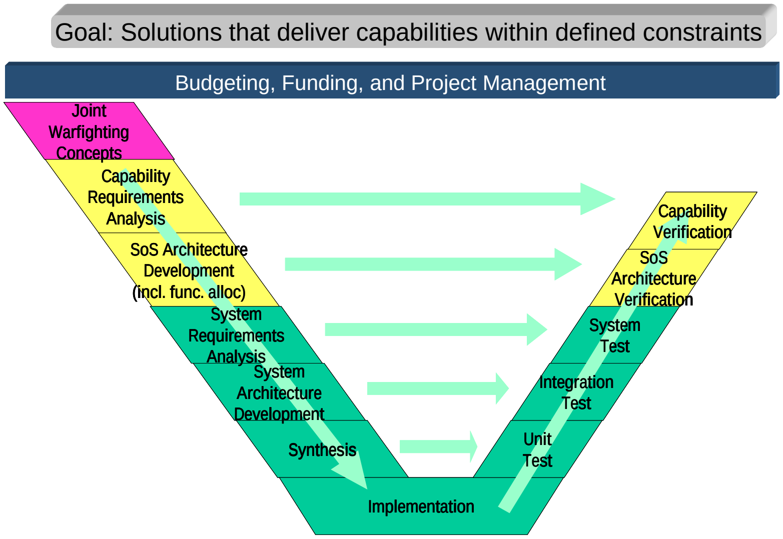
Why	**What**	**Who**	**How**
Architecture case studies	Architecture frameworks	Scope (Enterprise to	Architecture methods and

		Component)	processes
Architecture cost benefit analysis	Architecture viewpoints	Specialization (DOTMLPF)	Architecture project plans
Architecture research data	Architecture work products	Need (Consultant vs Practitioner)	Architecture tools

Table 1: Architecture of Enterprise Architecture Categories

WHERE IS ARCHITECTURE IN THE VALUE CHAIN?

Architecture methods, procedures, processes, plans, and tools aren't used in isolation. They are tied to an overall value chain. This section[3] presents an overview of the value chain as depicted in the following slightly modified INCOSE diagram.


Goal: Solutions that deliver capabilities within defined constraints
Budgeting, Funding, and Project Management
Joint Warfighting Concepts
Capability Requirements Analysis
SoS Architecture Development (incl. func. alloc)
System Requirements Analysis
System Architecture Development
Synthesis
Implementation
Unit Test
Integration Test
System Test
SoS Architecture Verification
Capability Verification

Systems Engineering Process with SoS Extensions

Figure 1: Notional System Engineering Process

Critical in the above value chain is the integration phase where multiple components, sub-assemblies, and assemblies come together to produce high-quality output on-time and within budget. However, it is actually not as simple as this as there is typically an integration chain where many components, sub-assemblies, and assemblies are developed concurrently and integrated at different levels – this can be referred to as a "solution integration value chain" that results in high-quality systems on-time and within budget, as depicted in the following figure.

[3] Please note that the example provided appears to be only relevant to systems or systems of systems, as opposed to en enterprise view, but since this systems view is more understood it provides a great opportunity to describe the relationship of architecture to other parts of the value chain. So although there may be a different scope in Enterprise Architecture, the following serves a purpose of juxtaposing architecture to engineering.

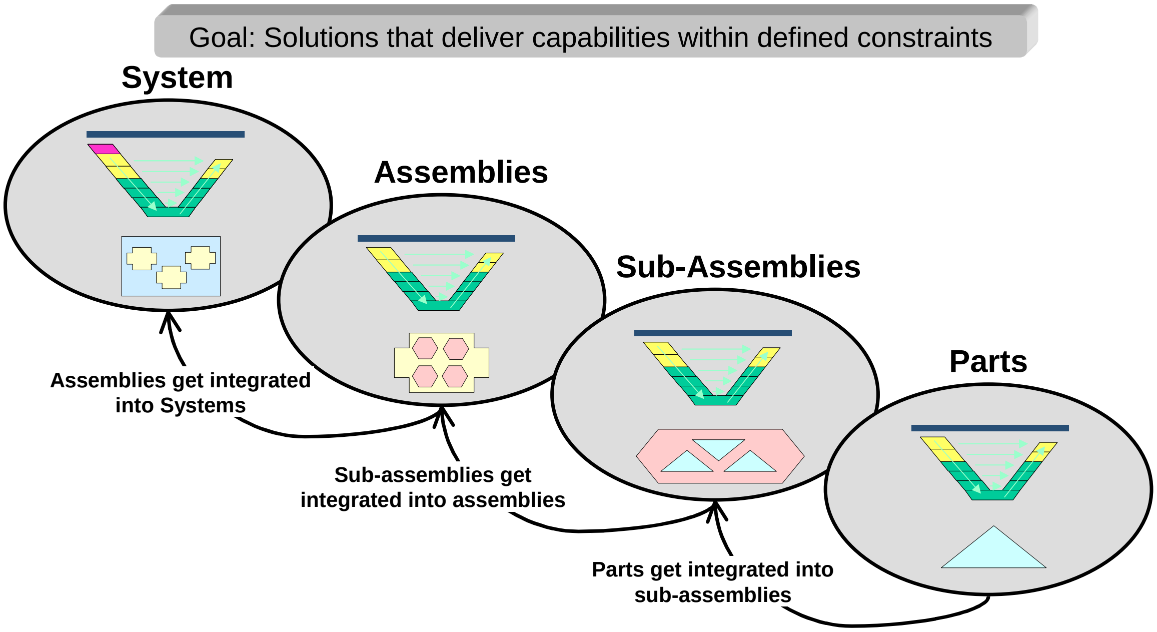

Goal: Solutions that deliver capabilities within defined constraints
System
Assemblies
Sub-Assemblies
Parts
Assemblies get integrated into Systems
Sub-assemblies get integrated into assemblies
Parts get integrated into sub-assemblies

Figure 2: A Classical Solution Integration Value Chain

A classical solution integration value chain yields a solution. As mentioned in the footnote early this example value chain is not necessarily in context to enterprise problems, but it does provide some insight into the position of architecture in the value chain and that position is:

- in the front end as it relates to understanding requirements
- the middle by inspiring or mandating subsequent design and engineering decisions to insure integration and interoperation, and
- the end by helping assess the resulting solution for fitness and subsequently inspiring change.

The processes that are part of this value chain fall into the general categories in the table below. The following table describes each of the processes in the value chain and highlights the architecture role.

Process Categories	Process Description and Output
Operations	Operations include execution of a mission where information technology is playing a role. Output of operations is successful execution of mission scenarios. For example win the war. **The architecture presents an abstraction of the components involved in the operational view.**
Program management	The process executed by the program manager to accomplish the tasking and meeting the objectives, schedule and fiscal constraints of the program. Must follow procedures and directives to manage a program from concept to deployment. Output is completion of a program. **The architecture represents a breakdown of the components of the solution and helps the program manager**

Process Categories	Process Description and Output
	decide on selecting the components that are fit for purpose. The architecture also represents requirements that are passed down to the solution providers.
Concept development	Concept development includes understanding the requirements (including quality), the goal, the costs, the overall look and feel, the style, the potential components suppliers, and the market of the product at a high level. Output is a conceptual design of the concept. **The architecture represents the earliest depiction of the components of the solution and helps the program manager decide on selecting the technologies that are fit for purpose.**
Definition/Solution Architecture	Definition/Solution Architecture includes translating the requirements (including quality), into a set of related components of a solution that deliver on those requirements and describes their relationships to each other, and to the environment, and the principles guiding its design and evolution. The output is the architecture of the system. The architecture should be materials in-specific, i.e. doesn't select the materials (wood, metal, .Net, Java, …) **The architecture represents a breakdown of the components of the solution and helps the program manager decide on selecting the components that are fit for purpose. The architecture also represents requirements that are passed down to the solution providers.**
Solution Design	Design addresses creating the engineering details to ensure that the final solution will meet specifications within quality and cost constraints. The output is a collection of detailed designs and an overall detailed model of the product. The designs are materials specific (wood, metal, .Net, Java, …). **The architecture represents the constraints that components of the solution must adhere to and the characteristics that will be assessed along the way.**
Development and Procurement	The development and procurement process includes managing resources, including inventory, human assets, machines, etc… to the end of producing components, parts, or the final system. The intermediate output of the development process is a collection of components and parts. The final output of the production process is a system, assembled from the

Process Categories	Process Description and Output
	components and/or parts, ready for final inspection. **<u>The architecture represents the constraints that components of the solution must adhere to and the characteristics that will be assessed along the way.</u>**
Integration	Integration is the act of bringing components of a solution together in an operational environment to ensure that desired capabilities are achieved. Integration applies to a system, a system of systems, and/or a family of systems. Integration includes quality conrol. The output of integration is a system, a system of systems, and/or a family of systems that delivers the needed capabilities in the environment in which it operates. This can sometimes include certification. **<u>The architecture represents the rules and constraints that components of the solution must adhere to to ensure effective and efficient integration and interoperation.</u>**
Deployment	The deployment process takes a system, a system of systems, and/or a family of systems and installs and configures it where ever it is necessary, e.g. all the operational theaters. The output of this process is operational system, a system of systems, and/or a family of systems in the operational theater. **<u>The architecture represents what should be tested and assessed as fit for purpose.</u>**

Table 2: Process Descriptions

The value chain is not just about architecture, nor engineering. It is broader. Architecture supports the entire value chain, beginning with working with the customer to understand real needs, as opposed to wants, and then throughout the process to translate those needs into capabilities verified to meet those needs. Additionally the architecture may be represented as different models to the customer that communicate how those needs may be met and it therefore plays an essential role in the consultative processes. First and foremost, it must be clear that the architecture must always be considered a representation of a solution for the customer, either an “as is” or a “to be” representation. Also we must always remember that architectures are only a means to effectively and efficiently get to a solution.

SOCIAL FRAMEWORK OF ARCHITECTURE

This section describes the various constituencies in the environment. The roles played by the different constituencies are not necessarily the same for every situation. The constituencies may take on different roles during the execution of processes, based upon local practice, organization, the budget allocation, or any number of other reasons. Therefore, in one operation a constituency may take a role that is taken by another constituency in another operation. For example, in one operation a contractor may do testing, in another operation the testing may be outsourced to specific testing organization. The roles adopted may actually change over time or per project.

The key organizations and entities in the typical environment relevant to the processes discussed above are illustrated in the following figure. Those organizations within the dotted circle are the primary constituencies. Those outside the circle are ancillary organizations that have peripheral involvement.

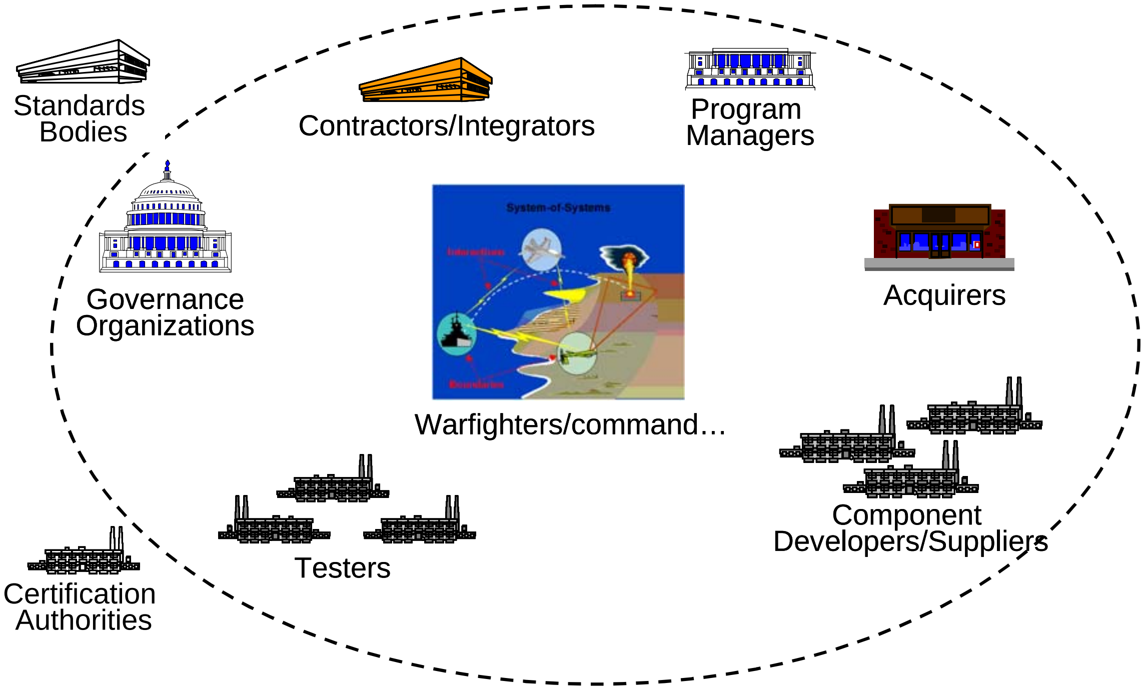


Figure 3: Architecture Communities of Interest

The table below describes the roles of these communities of interest and as well the challenges they face. Within the table key questions that require attention by the architect are highlighter. All of this is part of the social framework that the architect must understand.

Constituency	Roles Played and Questions for Architecture
Warfighters/command	The persons that are performing the mission, e.g. those fighting an enemy in the battlefield. Must be successful, keeping all resources capable with state of the art combat and combat support capabilities. **Can architectures help to create new capabilities by integrating technology on the fly? Can executable models help to understand the ramifications of doing on the fly integration, e.g. help understand what can be done to help win?**
Acquirers	The person or organization that procures a system, software product, or software service from a supplier. Must follow procedures and directives to acquire capabilities that meet the needs of the services. Acquirers engage in competitive tendering processes with contractors/integrators.

Constituency	Roles Played and Questions for Architecture
	Can architectures help those that acquire a solution to make better acquisition decisions?
Program Managers	The individual ultimately responsible for accomplishing the tasking and meeting the objectives, schedule and fiscal constraints of the program. Must follow procedures and directives to manage a program from concept to deployment. **Can architectures help program managers meet the objectives, schedule and fiscal constraints of a program by making better decisions?**
Contractors/Integrators	A company responsible for the end solution that goes to the customer. Deals with integration and final delivery. Must adhere to standards and deliver on contract. Contractors provide systems to services, typically used on large projects that deal with multiple vendors. Contractors engage in competitive tendering processes with acquirers. Contractors have alliances with vendors. **Can architectures help contractors remove ambiguity of requirements, including adherence to standards? Can architectures help contractors produce solution on time and within budget? Can architectures help contractors engage in the tendering process?**
Component Developers/Suppliers	A company/organization responsible for manufacturing a sub-assembly, such as a sensor, an engine, an instrument panel, a fuel cell, an application. Supply components typically as business partners to integrators; may be required to prove that their products meet certain criteria through certification. Must adhere to standards and deliver components and parts according to specification. Component developers have business partnerships with vendors and alliances with contractors. Engage in the standards and certification processes. **Can architectures help developers remove ambiguity of requirements, including adherence to standards? Can architectures help developers produce components on time and within budget? Can architectures help sell components?**
Testers	Organizations responsible for testing the quality of systems. Testers are spread throughout all of the above constituencies. Must objectively test/and verify that parts, components, and/or systems work as specified. **Can architectures help improve quality of solutions while not impacting costs and time?**

Constituency	Roles Played and Questions for Architecture
Governance Organizations	Organizations responsible for establishing and governing policy that may relate to executable models, this includes mandating standards such as 3170. Governance organizations impact standards bodies. Governance policy impacts purchasing, development, testing and program management organizations. **Can architectures help governance organizations by reducing ambiguity of regulations, or codifying a standard? Can architectures be used to improve governance itself?**
Standards Bodies	Organizations that develop technical standards and best practices that can establish the criteria for certification, e.g. 1471, 19760. Standards bodies are engaged in the standards process with vendors. Standards bodies have alliances with certification bodies. **Can architectures help standards organizations by reducing ambiguity of standards, or by codifying a standard?**
Certification Bodies	Organizations that provide certification and/or testing services, especially those involved with conformance certification and/or testing. Certification bodies have alliances with standards bodies. Certification bodies engage in the certification process with vendors and/or user organizations. **Can architectures help improve the stability of solution while not impacting costs and time?**

Table 3: Constituencies

The answers to these questions above are each yes, but how represents the challenge for all architecture efforts. The architect community must strive to answer these questions and develop repeatable methods to effectively and efficiently provide the answers.

- Can architectures help to create new capabilities by integrating technology on the fly? Can executable models help to understand the ramifications of on the fly integration, help understand what can be done to help win?
  - Yes – by driving requirements for “ready for integration and/or interoperation”
- Can architectures help contractors and developers engage in the tendering process, produce solutions on time and within budget, remove ambiguity of requirements, including adherence to standards?
  - Yes – by ensure that requirements are well articulated and embedded consistently within the architecture.
- Can architectures help governance organizations by reducing ambiguity of regulations, or codifying a standard?

    - Yes – by ensure that requirements for standards and profiles of standards are well articulated and embedded consistently within the architecture.

- Can architectures help improve quality and stability of solutions while not impacting costs and time?
    - Yes – through tradeoff analysis.
- Can architectures help program managers meet the objectives, schedule and fiscal constraints of a program by making better acquisition decisions?
    - Yes – by clearly depicting the components of a solution and the roles and rules that those components must adhere to.
- Can architectures help sell components?
    - Yes – by concisely depicting the components of a solution, its role, and the value of that component.
- Can architectures help standards organizations by reducing ambiguity of standards, or by codifying a standard?
    - Yes – by ensure that requirements for standards and profiles of standards are well articulated and embedded consistently within the architecture.

The “so what” here is that the architect must understand these different perspectives, and create architectures that can address these questions, but do so in a manner that is understandable by each of the constituencies. The architect is the consummate broker, between the customers and the developers of solutions and as such must understand and use the different languages based on the community.

ARCHITECTURE LEADERSHIP

This section on “Architecture Leadership” provides additional insights on the characteristics of leaders in the architecture discipline. The characteristics apparent in those architects that have successfully used and sold their architecture efforts are presented here.

Ten characteristics of the ideal systems architect are captured very well by Diane P Hanf which was inspired by reading TWELVE SYSTEMS ENGINEERING ROLES, Sarah A. Sheard, Software Productivity Consortium and ARCHITECTING LARGE BUSINESS SYSTEMS, TUTORIAL AT OOP 2001, MUNICH.

1. Excellent Communicator – Is able to articulate concepts at varying levels of abstraction (simple and detailed), both verbally and nonverbally, but always elegantly, to diverse audiences.
2. Inductive Analyst – Is able to produce the “so what” based on observation of the environment, trends, and knowledge of what has worked in the past.
3. Experienced in the System or Domain (System Domain Knowledge) – Knows the items being architected; understands interrelations among systems in the domain.
4. Business Savvy – Understands the economic & schedule language and implications of acquiring systems and its impacts to a solution architecture to open a line of communications with a program/project manager and express the business aspects in the architecture.

5. Broad Systems Discipline(Technology) Knowledge – Understands each physical discipline technology and strength in enough depth to represent it accurately at the solution architecture level.
6. Great Detective – Is able to discover and extract not-so-obvious meaning and impacts of various aspects of an architecture so that it can be articulated and examined.
7. Good Judgment & Thorough – Is able to make sound decisions about the architecture based on experience rather than mathematical calculations and ensures the integrity & balance of the architecture is reflected at each lower level.
8. Operationally (Business Process or Corporate) Savvy – Understands the business environment—processes and language-- to communicate with operators and users and provide traceability and accountability for the requirements throughout a solution architecture.
9. Visionary – Sees beyond the current boundaries of a solution and can easily predict logical paths of progression for that solution and shows how a solution might get there by articulating in architecture views.
10. Knows the Codes – Understands the regulations, technical and other standards that bound the solution architecture enough to express it and come up with solution configurations that fall within the governance framework.

Eugene Pierce suggests that one consider leadership styles presented in an article by B. Kim Barnes entitled “Leadership in Turbulent Times.” In this paper the author presents a few leadership styles two of which seem to map well to the needs of the architect. These are the “Peaceful Leader” and the “Fascinating Leader.”

The behavior of the Peaceful Leader includes:

“The Peaceful Leader:

- Asks open-ended questions
    - “How do you think we can solve that problem? What ideas do you have?”
- Draws out
    - “Tell me more about your idea. How do you see that working?”
- Checks understanding
    - “So you are willing to negotiate with the other team on that...”
- Encourages
    - “I am confident that you will succeed as you have done before with tough issues like this.” “

The behaviors of the Fascinating Leader includes:

“The Fascinating Leader:

- Envisions
    - “I can see us operating like a world-class relay team, bringing the project in on time and winning the race to market.”

- Offers incentives
    - “If you are willing to be part of this effort, I will make sure you have all the help you need”.
- Tests implications
    - “You haven’t said so, but your comments indicate that you might be thinking about a way you can contribute to this effort.”
- Identifies with other
    - “If I were you, I’d be both excited and apprehensive right now—we are starting something so important to the future of this company.” ”

ARCHITECTURE SKILLS

To complement the subject of Architecture Leadership the following sub-section on required architecture skills provides architects with a list of skills, that when honed, best prepare an architect for the task at hand.

The following is a laundry list of skills. Follow-up analysis can help us hone this list.

- Architecture Process and Method Use
- Strategic Planning Methods
- Architectural Styles of Computing Use
- Listening, Questioning and Probing Techniques
- Problem Solving and Analysis Techniques
- Teaming/Working with Others/Networking (remotely)
- Visual Modeling Techniques
- Written and verbal Communication
- Negotiation and Conflict Management
- Synthesis (Aggregation and Pattern Matching Techniques and Abstraction)

The following represents knowledge that is very useful for the architect:

- Related enterprise processes (in DoD JCIDS, PPBE, D-5000.2)
- IT Trends and Directions
- System/Technology Integration
- Domain Specific Awareness
- Modeling frameworks (Zachman, TOGAF, FEA, …)

ARCHITECTURE ROLES AND RESPONSIBILITIES

Architects are responsible for the following.

Create useful models - Take the requirements and develop well formulated conceptual and logical models of the elements of the problem and elements of the solution, augmenting the models as necessary to fit all of the circumstances. Show multiple views of the models to communicate the ideas effectively to specific stakeholders such as stakeholders involved in operations, stakeholders involved in managing the environment, and stakeholders involved in developing the environment. These models should be assessed in terms of their usefulness by how well they support the acquisition process. These models should also be assessed in terms of their effective and efficient reuse within the entire process, for example directly reusable by developers if development is warranted, directly reusable by testers. Over time models will become more sophisticated; the architect needs to stay current with respect to new modeling techniques and tools and should use those new techniques to ensure that the appropriate solutions are delivered.

The architect is responsible for the overall **architecture integrity** and maintaining the vision of the offering from an architectural perspective. The architect also ensures leverage opportunities are identified, using building blocks, and is a liaison between the participating organizations (especially development and functional organizations) to ensure that the leverage opportunities are realized. The architect provides and maintains these models as a framework for understanding the domain(s) of development work, guiding what should be done by specific COTS vendors or contractors. The architect must represent the operational view of the architecture by understanding all the necessary business components. The models created by the architect must adhere to the guidelines established by regulation and must ensure that the architecture models are appropriately positioned with other architectures

Understand and interpret requirements - Probe for information, listen to information, influence people, facilitate consensus, synthesize and translate ideas into actionable requirements, articulate those ideas to others. Identify use or purpose, constraints, risks, etc. The architect participates in the discovery and documentation of the customer's business scenarios that are driving the solution. The architect is responsible for requirements understanding and embodies that requirements understanding in the architecture specifications. In addition to understanding the "customer" side of requirements, the architecture must balance those requirements with program requirements, such as affordability and timing, and enterprise requirements such as fitting into the existing environment and ensuring interoperability with enterprise applications and integrating into the enterprise infrastructure. All of this together creates a more realistic understanding of what it takes to deliver a solution.

Validate, refine and expand the models - Verify assumptions, bringing in Subject Matter Experts, etc. in order to improve the models and to further define them, adding any necessary new ideas to make the result more flexible and more tightly linked to current and expected requirements. The architects additionally should assess the value of solution-enhancing developments emanating from industry and incorporate these into the architecture models as appropriate.

Manage the architecture - Continuously monitor the drivers of the models and update as necessary to show changes, additions and alterations. Represent architecture and issues during decision points of the program. The architect is an "agent of change," representing that need for the implementation of the architecture. Through the development cycle the architect continuously fosters the sharing of customer, architecture and technical information between organizations. The architecture must manage the architecture artifacts under change control and ensure that the artifacts are accessible.

Represent the architecture – Continuously maintain an architecture vision that is convincing and acceptable to customers, developers, and decision makers. The architect continuously communicates the content and options expressed in the architecture. Also the architecture communicates the merits expressed in the architecture based on objective analysis weighing subjects like needs, costs, viability, and timing.

Under most circumstances, the complexity of a solution requires multiple architects to support the architecture effort. There are people with architecture responsibility in the Functional Organizations, in the System Program Offices, in organizations that provide infrastructure, and within the System Integrators. In order to address the requirements, architecture work is required from each of these organizations; the Functional organization must create the operational view, Systems Integrators must create Systems and Technical views for infrastructure and solutions, and these architectures must comply with foundation architectures created by governance bodies. Any of the architects involved in a program perform the tasks described above. In cases where a team of architects is deemed necessary, a lead architect should be assigned to manage and lead the team members. In this document the lead architect is identified as the "Chief Architect."

REFERENCES

[To be supplied]

BIOGRAPHY

Terence Blevins is a MITRE employee matrixed to the Air Force in the role of Branch Chief and Lead Architect of the Air Force Operational Support Enterprise Architecture. Mr. Blevins has been involved with the architecture discipline since the 80s when he was at the NCR Corporation. He has been involved with evolving this discipline since 1996 when he first was introduced to The Open Group's Architecture Forum. He was co-chair of the Architecture Forum and frequent contributor of content to TOGAF including the Business Scenario Method.

Mr. Blevins holds undergraduate and Masters degrees in Mathematics from Youngstown State University. He is TOGAF 8 certified.