A Management Guide to Software Maintenance in COTS-Based Systems

May 1998

Judith A. Clapp
Audrey E. Taub

MITRE
Center for Air Force C2 Systems
Bedford, Massachusetts

Abstract

The objective of this guidebook is to provide planning information that results in cost-effective strategies for maintaining Commercial Off-the-Shelf (COTS) software products in COTS-based systems. It considers the issues and risks in using COTS software over the life cycle and how to control them. It describes changes in the software maintenance process that are needed to manage a COTS-based system. It provides guidance in developing a COTS Software Life-Cycle Management Plan.

KEYWORDS: COTS software, software maintenance, COTS-based system, life-cycle planning, sustainment

Table of Contents

List of Tables

Section 1

Introduction

1.1 Objective

The objective of this guidebook is twofold: (1) to provide planning guidance that results in low risk and cost-effective strategies for maintaining Commercial Off-the-Shelf (COTS) software products in COTS-based systems, and (2) to provide guidance on the preparation of a COTS Software Life-Cycle Management Plan.

1.2 Rationale

The functional requirements for systems in the commercial world have moved closer to the requirements for military systems. For example, electronic commerce and military systems both require network security. This has led to more extensive use of COTS products, including hardware and software, in military systems. The DOD, in policy and in practice, has stated a strong preference for COTS products in systems and for a reduction in custom-built software or hardware. However, the use of COTS products can significantly change the process by which systems are maintained in their operational phase. We are just beginning to experience and understand these changes, and to recognize that life-cycle planning for COTS-based systems must take into account, in early planning, the issues that must be confronted in order to facilitate the maintenance phase. With the use of evolutionary acquisition and the spiral model, the operation and maintenance phase starts sooner and continues for a much longer portion of a system's life cycle. This makes early life-cycle planning for maintenance of COTS products even more important.

Much has been written about the selection and use of COTS hardware products in system development. Less has been written about the use of COTS software products. Even less has been written about the maintenance of COTS software products in systems. With COTS-based systems, maintenance planning must control risks associated with rapid turnover of software products and the impact of these changes on the rest of the system. There are technical, administrative, and financial issues to decide. In addition to new kinds of risks, such as loss of vendor support and lack of source code, COTS products open new opportunities for the program managers to transition their maintenance responsibilities to contractors, to introduce new technologies, and to acquire tools that automate some of the maintenance of these products.

1.3 Approach

It is our contention that there is no single way to manage the sustainment of all COTS-based systems. Rather, the approach for each system will depend on the operational and technical characteristics of the system, the administrative policies and constraints placed on the

program, and its financial situation. Our approach to this task was to develop a generic COTS life-cycle plan and guidelines that show how it can be tailored for each system depending on its specific requirements and constraints.

The basis for our guidance has been a review of DOD, MITRE, and industry experiences and lessons learned in COTS product applications, and attendance at meetings/workshops whose focus was on the implementation of COTS-based systems.

Section 2

Introduction to COTS Products

In this section, we explain the different ways that the term "COTS" has been used, the role COTS products can play in a COTS-based system, and why maintenance of COTS-based systems is different from custom-built systems.

2.1 What are COTS Products?

The term "COTS" has been used to label many different kinds of software products. The Federal Acquisition Regulations (FAR) define a "commercial item" in several ways. The basic definition is:

> Any item that is customarily used for nongovernmental purposes and that has been sold, leased, or licensed to the general public, or has been offered for sale, lease, or license to the general public.

Note that this definition does not appear to require that the general public has actually acquired the item, only that the item has been *offered.* It does mention that the item should have *nongovernmental* purposes, which implies that there will be a larger market than the government for the product.

The FAR goes on to offer variations based on this definition, such as:

- An item that is evolved from one meeting the prior criteria that is not now available in the commercial market, but will be available in time to meet delivery requirements.
- An item that, except for minor modifications to meet DOD requirements, satisfies the prior requirements.

In order to gain the advantages of COTS products, we believe that COTS software products should be defined as those that are offered to the public and are actually used by the public in the same version as those in military applications. In other words,

- The products should have a set of users in the non-military community.
- The products should be offered via sale, lease, license, or "freeware" or "shareware" (no cost) to the public.
- The products should not be modified for military use, except in ways that the vendor intended any user to modify them, although there may be minor modifications for the DOD that will then be offered to the public.

COTS software products can include associated proprietary tools for modifying the behavior of the products and their outputs. These may be visual tools for defining a graphical user interface, or macros or scripts for extending or tailoring the software. In contrast to these

tools, a system may consist of "custom" software that is developed in a programming language and compiled to perform the functions that are not covered by any COTS product.

If a COTS software product has been internally modified in ways not normally available to the commercial user, then it will be considered a "modified COTS product" unless the modifications become available commercially as part of a later version of the product. Modified COTS software products will require careful coordination with the vendor or whoever makes the modifications to determine how the modified versions will be maintained in the system and who will do that maintenance.

In this report, the term "COTS product" most often refers to COTS software products, although there are cases where what is said applies equally or explicitly to COTS hardware.

2.2 What are COTS-based Systems?

A COTS-based system is one that contains components that are COTS products. These components may perform generic functions that are independent of the system's application domain. Usually, these are part of the infrastructure of the system. The Common Operating Environment (COE) is an example of such a collection of COTS products. The components include the operating system, the database system, networking, and displays. Other components of the system may be very dependent on the services of these components.

Incorporating COTS software products to form a COTS-based system involves designing their interfaces with other system components, and the work of integrating them. Integration can be a substantial effort. It depends on generating what is commonly referred to as "glue" or "glueware," which may be in a mixture of languages including those associated with each product. The number of interfaces COTS product has with other system components, and the number of components it interfaces with, determine the extent of integration. The interface that a COTS software product offers to programmers is called the Application Program Interface (API). When its definition is available and follows standards, it is called an "open" rather than a proprietary API. For example, there is a Standard Query Language (SQL) that can be used as an interface for extracting information from a database management system. That is an open interface.

In managing COTS-based systems, it is important to include not only the COTS products that become part of the operational system, but also the COTS products in the software development environment, and products that document the system requirements, and manage the maintenance of the system. The line between development products and operational system products has become blurred by development tools such as Graphical User Interface (GUI) builders that generate the user interface software that becomes part of a system.

2.3 How is the COTS Software Maintenance Process Different?

"Software maintenance" here refers to the set of activities that are performed to keep a system operational as software changes after the system has been fielded. Software

maintenance begins as soon as a system has been released to users for the first time in the case of incremental, evolutionary, or spiral developments. It then encompasses modifications to subsequent releases of the system. At the time of software maintenance, a set of system components are in place, the system has been tested and accepted for operational use, operators have been trained, and logistics support has been arranged. Data accumulates in system databases as a result of operational use.

The maintenance of software for COTS-based systems has different risks from those of custom-designed systems and requires different maintenance activities. The risks will differ for each system depending on operational characteristics of the system, design characteristics, and the choice of COTS products. The maintenance of COTS products in a system also requires a change in the usual maintenance activities. Some of the important differences in risks and in the process of maintenance for COTS-based systems are discussed below.

2.3.1 Risks

The following is a list of some of the significant risks that must be controlled before or during the maintenance of COTS products in a system. A COTS Life-Cycle Management Plan must address these risks.

2.3.1.1 Program Management Control over Vendor Product Releases

One of the most difficult problems in maintenance of COTS-based systems is the decision to upgrade or replace a COTS product. The program management generally cannot control the frequency or the content of new COTS releases. The timing of a new COTS product release tends to be asynchronous with and independent of the new releases of other COTS products and components in the system. Changes in the operational requirements for the system are not always easy to synchronize with product upgrades either.

New releases of COTS software products can occur as often as every six months. Failing to upgrade to the latest version can result in:

- Loss of vendor support for prior versions that are installed and in use.
- The inability to buy new copies or obtain licenses for additional copies of the version that is already in the system. This is necessary when the system is being installed in sites incrementally over a period of time.

Upgrading to the latest version can result in risks such as the following:

- The new software version is incompatible with other COTS software products in the system, necessitating updating of those products too.
- The new version has new data formats that require changes to be made to the formats and contents of existing files and databases that were created by prior versions of the COTS software.

- The new version requires changes to the tailoring, adapting, and integrating that had been done for a COTS product if the new release makes changes in the tools and languages that are specialized or proprietary to that product.
- The new version of the software is incompatible with the version of the hardware that is in the system.
- A new version of the hardware is introduced into the system that forces changes to the existing versions of the software to make them compatible or because timing has changed under the new hardware.
- The new version of the software changes the user interface in ways that require retraining operators.
- Changes in the consumption of time or memory resources by upgrades to COTS software are not compatible with the system requirements or the hardware capacities.
- The new version forces changes in the operational capabilities of the system because it no longer supports those capabilities in the same way or at all.
- A new version may provide more capabilities that may have to be suppressed or restricted due to security concerns.

2.3.1.2 Life-cycle Planning and Budgeting for COTS-based Systems

The funding profile over the life cycle of a COTS-based system is different from that of custom-developed software systems. According to a recent draft of an ASD (C3I) memorandum entitled *Life Cycle Consequence of a COTS Acquisition Strategy*:

> "COTS spreads the funding requirement more uniformly across the program life cycle. The understanding and management support for significantly greater sustainment cost is not evident in program guidance and budget direction. The immediate issue which confronts the Services and Agencies is that information policy and the derivative budget guidance has not been revised to account for these consequences."

A large source of funding risk is due to the uncertainty about how often COTS software components will have to be upgraded and even replaced, and how much more of the system may have to be changed as a result. The same uncertainty holds true for hardware that may be obsolete or may be made obsolete by a change in the COTS software.

Use of COTS software is financed through licensing over the life of its use. This cost can be substantial compared to the initial cost for development. There can be wide variations in the kinds of licensing agreements that are made, which in turn lead to wide variances in the cost of using COTS software. This adds to the difficulty of predicting software costs over the life cycle of a system and necessitates good business acumen in negotiating and renegotiating licensing agreements. Often negotiating may be most economical when it is across systems, which can require a third party to make the arrangements.

2.3.1.3 Controlling Security of the System and the Support Environment

Maintaining security of the software and the data can raise a number of issues with military systems that contain COTS products, e.g.:

- The facilities in which the system operates may be classified, the code may be classified, or the data may be classified. This may necessitate use of cleared personnel for maintaining and debugging COTS software. Vendors or developers of COTS software may not be able to supply cleared personnel.
- Even more important is the risk that the COTS products may introduce a security vulnerability into the system, since neither the designers of the products nor the design and implementation are visible to the government. This may demand special procedures for testing or certifying that the COTS products do not present security risks. Each time the COTS products are changed, the system may have to be recertified.
- Deployable systems may require maintenance personnel who can relocate quickly to a hostile environment that is inappropriate for civilians. DOD policies may preclude the use of civilians with deployed systems. In this case, military maintainers must have sufficient information to maintain COTS products in a deployed environment. For software, this must be done without source code. According to AF Executive Guidance, January 1996, "The AF must structure logistics manpower to ensure direct combat support functions at the theater and tactical levels of conflict are maintained as 'blue-suit' operations."
- There are versions of software that are not for export, especially when encryption capabilities are involved. This may restrict their use in systems that will have foreign users.

2.3.1.4 Maintaining Software Quality

Maintaining software quality involves, among other things, identifying the source of errors, repairing them, and assuring that the system is free of errors. Since COTS products are "black boxes," the maintainers of COTS-based systems will not have access to the source code. This introduces the following risks:

- The quality of the software that is delivered for a COTS product depends on the supplier's view of quality. For many vendors, the competition for rushing a new version to market is more important than delivering a high level of software reliability, usability, and other qualities.
- It can be very difficult to locate the source of errors when there are multiple COTS products in a system. The problem may lie in some interaction between or among COTS products, or between one or more COTS products and custom software. The difficulty in locating and fixing errors is compounded when the products are from different vendors.
- Vendors or developers of the COTS software may have to be involved both in debugging software and in making the repairs. This means that the response time for repairs may be determined by the vendor and not the program management. This

problem may be exacerbated if the military system uses a COTS product in a way that is different from the vast majority of commercial users, so the vendor has little incentive to make the correction. This lack of response, in turn, increases the risk that the program manager has to change products or use a unique version of a COTS product.

- When a new version of a COTS product is integrated into a system, it must be tested. Often, the product contains new features that are not used by the system. There is a risk that these unused features may still have some indirect impact on the behavior of the system. This risk can complicate testing if the COTS software must be tested for more than those functions or features that are directly used.

2.3.1.5 Configuration Management

Configuration management becomes a risk for COTS-based systems for two reasons. First, new versions of components may have to be installed frequently. Second, it may be easy for maintainers at different sites of one system to obtain replacements or upgrades for COTS products directly from vendors without following configuration management procedures.

2.3.2 Maintenance Activities

The previous list of risks associated with the use of COTS software products can be controlled by a set of interrelated activities that should be included in the maintenance of a COTS-based system. These are either additional activities or modifications to the usual maintenance activities. The COTS Life-Cycle Management Plan for a given program should show how these activities will be performed and which organization will perform them.

As noted earlier, the biggest differences in the process of maintenance for COTS-based systems are:

- The initiative for change in versions of COTS components is by the vendors and not planned by the program management.
- The changes introduced by new COTS products or releases are not in response to change specifications by the program management to meet new requirements or correct problems.

Consequently, the program management has no control over when changes appear and what they are. The changes may result in a loss of capabilities that the system depends on. The addition of new or improved capabilities in a new version may offer opportunities to improve the system's capabilities or to overcome existing shortcomings.

The following are brief descriptions of new or modified software maintenance activities for COTS-based systems. More detailed guidance for each activity will be given in Section 3.

2.3.2.1 New Maintenance Activities

The following are new activities that are performed during software maintenance to help control risks due to the use of COTS software in a system.

- **Market research**

  Market research should be conducted regularly during software maintenance to determine the availability of new commercial hardware and software products and to estimate how soon and what kinds of changes will be made to the COTS products that are already in a COTS-based system. The information obtained from market research is used to perform an impact analysis for any replacements being considered for COTS software products or hardware products that may affect the software product configurations.

- **Upgrade impact analysis**

  This is the activity to determine the cost and difficulty of introducing a replacement for a COTS product that is either a new product or a newer version of the existing product. This activity must consider all of the ways that the product might cause changes to the operational system, the software in the system, or the software maintenance process.

- **Asset management**

  International Data Corporation defines asset management as "a collection of tools, services, and financing options that enable users to purchase, operate, and dispose of computer products at a predictable life-cycle cost." [IDCNet bulletin #13253, March 1997.] In other words, it is the management of the financial aspects of acquiring, using, servicing, and upgrading COTS products. For software, asset management focuses on an inventory of COTS products, versions, where they reside, and financial obligations such as licenses associated with them.

2.3.2.2 Modified Maintenance Activities

The following are maintenance activities that are modified to help control risks due to the use of COTS software in a system. Each of them is related to the risks described earlier. More details can be found in Section 3.

- **Program planning and budgeting**

  This activity must include the additional costs for periodic renewal of licenses or upgrades of COTS software versions.

- **Quality control**

  This activity must include the debugging, repair, and testing of installed and proposed COTS software. It also includes the insertion into the system of fixes that a vendor will deliver for a released version of a system.

- **Configuration management**

  This activity must include tracking and controlling the versions of all COTS products and custom software installed at all locations in the system.

- **Logistics support**

  This activity includes the acquisition, distribution, and installation of copies of upgraded COTS software products. It also includes the re-installation of COTS software products when the installed copies have been corrupted, which is not uncommon for some COTS software after a period of use.

Table 2-1 shows the risks that are addressed by each of these COTS-related maintenance activities.

Table 2-1. COTS-Based Risk Management Activities

Risk	Risk Management Activity
Control over vendor product releases	Market research Upgrade impact analysis
Life-cycle planning and budgeting	Asset management
Controlling security	Quality control Logistics support
Maintaining software quality	Quality control Asset management Logistics support
Configuration management	Asset management Logistics support

Section 3

Guidance for a COTS Software Life-Cycle Management Plan

This section contains guidance in making the decisions that determine a COTS Life-Cycle Management Plan. It contains general guidance on the decisions that must be made during the planning and implementation stages that affect COTS software maintenance, and more specific guidance on the maintenance activities related to COTS software. It is important to recognize that the activities listed are interdependent. When decisions are made for any one activity, the effect on the other activities must be considered.

3.1 Major Decisions

The following are decisions that affect the management of COTS products during the life cycle of a system:

- Select COTS products using criteria that include maintainability for the system and the products.
- Determine who will acquire the COTS products, and the conditions for their purchase or license with consideration for their effect on maintenance costs and services.
- Determine the organization(s) that will be responsible for each activity in the COTS product maintenance process (listed above in Section 2.3.2).
- Define a relationship with the product vendor (developer) if not the primary maintainer.
- Determine how training will be provided for COTS product maintenance if any organic support is required.
- Select and acquire tools to support COTS product maintenance.
- Determine procedures to be used to support COTS product maintenance.
- Determine support strategy for upgrading and replacing COTS products.

All of these decisions should be documented in a COTS Software Life-Cycle Management Plan, as described below.

3.2 Preparing a COTS Software Life-Cycle Management Plan

Table 3-1 is a sample plan outline for documenting the major decisions previously listed and any other decisions that affect the maintenance of COTS software in a system. It can be tailored for the needs of an individual program. Note that the plan is a life-cycle plan to emphasize that it must cover all maintenance considerations throughout the life cycle of the system, not just during maintenance.

Table 3-1. Sample Outline for a COTS Software Life-Cycle Management Plan

1.0 Introduction

Definition of terms - what is meant by "COTS," and other terms specific to this plan.

2.0 Program Requirements and Constraints

Requirements and constraints on the use of COTS products that were imposed on the program by organizations outside the Program Office

3.0 Preparing for COTS Software Maintenance

Activities and decisions prior to system maintenance that affect COTS software maintenance

- **3.1 COTS Product Evaluation Criteria**
- **3.2 COTS Product Selection**
- **3.3 Purchasing and Licensing Arrangements**
- **3.4 Organization and Responsibilities for Software Maintenance**

4.0 COTS-Based Software Maintenance Procedures

Procedures for activities performed during maintenance of the COTS-based system that affect COTS software product maintenance

- **4.1 Market Research**
- **4.2 Upgrade Impact Analysis**
- **4.3 Asset Management**
- **4.4 Program Planning and Budgeting**
- **4.5 Quality Control**
- **4.6 Configuration Management**
- **4.7 Logistics Support**

What follows is guidance on making decisions in preparing the life-cycle plan for a specific system. The actual process and contents of the plan will depend on what stage the program is in when the plan is made. For example, a program in the early stages of planning can record the externally imposed constraints on the use of COTS products, but many choices may still be possible in selecting products, organizing for maintenance and determining maintenance procedures. If a system has already been designed, and COTS products purchased, then there are many more constraints on how COTS software maintenance can be done. The plan should document these *a priori* decisions as well as the results of subsequent decisions.

3.3 Program Requirements and Constraints

Before proceeding with a plan, it is important to note the goals for the use of COTS products and the policies, rules, regulations, and previous decisions that will limit the options on what products are chosen or how they will be managed during software maintenance. These requirements should be documented in Section 2 of the plan if they were imposed by organizations external to the Program Office. The assumption is that the Program Office had little or no influence over these decisions. For example, the system may be required to use COTS products that conform to a set of common specifications such as the Defense Information Infrastructure (DII) Common Operating Environment (COE). Other constraints might be due to interoperability requirements with other systems, constraints on the choice of computer hardware, constraints on what organization will provide the software, or a requirement to use a specific system maintenance organization.

Sometimes the constraint is a general policy of the DOD or acquisition organization to maximize the use of COTS products. If a developer has proposed one or more COTS products, then the proposal ought to justify the choices by indicating the perceived benefits in cost, schedule, and system performance. The constraints imposed by the choices ought to be documented as well. For example, the choice of a COTS product may limit the design of the rest of the system or may preclude use of other products.

3.4 Preparing for COTS Software Maintenance

These are the kinds of activities and decisions made before the start of system maintenance that will influence how software maintenance is done and what it will cost. Before the decisions are made, the procedures for making them can be documented in Section 3 of the COTS Software Life-Cycle Management Plan. As these decisions are made, they should be added to Section 3 of the plan.

3.4.1 Establishing COTS Product Evaluation Criteria

The criteria will be a function of program constraints, system operational requirements, system design requirements, and maintenance concepts including logistics. They should be documented in the plan so they can be used by maintainers to understand the original choice of products and for selecting replacement products. Here are some examples of criteria that impact the ease of maintenance, especially making modifications to the system.

- Does the product have a wide customer base, so its usability and quality have been and will continue to be tested and improved, and it is likely to stay on the market?
- Has the product been on the market long enough to be stable and reliable in operation? What is its reputation for quality?
- Does the vendor provide useful documentation and a help desk, and respond quickly to problems?

- Does the product use standard or proprietary interfaces (open or closed)? This affects how readily other COTS products may be available to interface with it, and how difficult it is to integrate custom software and other COTS products with it.
- Does the product adhere to standards or have variations or extensions? Are the standards stable?
- Has it been integrated with other products planned for this system? If so, what kinds of problems were encountered?
- How easily is the product adapted or tailored, what is the method for doing it, and how easily can it be learned?
- How frequently have upgrades been made to this product? How soon is the next upgrade likely to appear?
- How compatible have upgrades of this product been with each other? What changes did they cause?
- How many different platforms and operating systems has this product been implemented for? Does it get upgraded when new versions of operating systems are released?

3.4.2 Selecting COTS Products

Selecting COTS products requires an evaluation process as well as a set of criteria. The procedure by which COTS products will be selected should be documented in the plan. For example, will the products be selected through a prototype phase, market research, experimentation, or a fly-off competition? These procedures or equivalent ones will also be needed during maintenance to evaluate the upgrades to software products as well as new products before deciding to introduce them into the system.

One consideration in the evaluation process is how COTS products will be obtained for comparison and evaluation. It can become expensive if all candidates being considered have to be purchased, especially for expensive software such as network managers or database management systems. Some vendors will provide an evaluation copy at no charge for a short period of time, or a demonstration copy with a very limited capability to try. It can be advantageous to obtain a "90-day return" clause in a license for a product in order to try integrating it with an existing system. Computer facilities and test drivers will be needed much earlier in the life cycle when COTS products are being evaluated by implementing and executing a prototype or some portion of the system with the product.

Decisions on selection must consider the total set of COTS products as well as the merits of individual products. An important decision about the choice of COTS products that can have a large impact on maintenance is the number of vendors represented by the selection. Clearly, the more vendors, the more likely it may be that incompatibilities will arise between vendors' products when upgrades are installed, and the more likely that the products will have independent upgrade schedules.

Once the products have been selected, an inventory of the COTS products should be maintained throughout the life cycle. A description of a database to maintain that information is given later in this guide.

3.4.3 Deciding on Purchasing and Licensing Arrangements

The major decisions concerning COTS software licenses are:

- Who will buy the licenses?
- Who will own the licenses?
- Who will maintain the licenses?
- Who will renew and pay for the licenses?
- What conditions the should licenses contain?
- When to acquire the software and licenses

These decisions affect the total cost of the licenses and their renewal as well as the flexibility in how and where the software is used.

The general rule on buying COTS software is to be part of as large a buy as possible to get volume purchase discounts. Possible commercial sources are software license resellers who buy in volume and resell smaller quantities at a discount. The contractor who is developing or will be maintaining the system may be such a reseller. DOD organizations, such as DISA, have been able to license or buy large quantities of COTS software and distribute them to individual programs. The availability of other DOD contracts for buying in quantity and redistributing software should be checked because it changes over time. Some DOD contracts provide products only and others provide services as well.

The buyer of the software can be different from the owner of the license, the maintainer of license information, and the organization that pays the annual fee for the license renewal. The license is granted to an owner, who will receive services granted under the license. It may not be possible to change owners without changing licenses. This can affect maintenance of a system if the maintenance organization can change over time, e.g., from the development contractor during concurrent development and maintenance to a different organization after some period of time. Maintaining the license and renewing it will be discussed in the section on software maintenance activities. It is important to make clear which organization will perform these activities at the time the licenses are obtained. Options include the owner of the hardware in systems such as MAMS, where the clients all belong to the users while the servers are owned by the MAMS system. Another option might be a common logistics support contract for one or more systems.

There are many different options that will determine the cost for a license. These are some of the conditions that the options on a software license can determine:

- What version of software is being licensed

- Who may use the software
- On what hardware
- When
- For how long
- In what location
- How usage will be monitored
- How many copies may be distributed
- What documentation is provided and in what form
- What level of help and maintenance services will be provided
- Whether changes can be made to the version of software, owner of the license, or any other terms above.

The choice of options needs to be made with a life-cycle view of how the software will be used, how the system will be maintained, and what the impact will be on long-term costs. Often the operational concept and design for the system must be known in order to determine the best conditions for licensing. The options that determine the basis for the cost of a license can be very complex, especially if the system has multiple users distributed in a client-server environment. For example, if the license cost is determined by the number of unique users by name, then a three-shift operation would have a much higher cost on this basis than if the cost is determined by the number of clients computers. Similarly, if there is low activity in a system, it might be advantageous to get a license whose cost is based on the maximum number of simultaneous users as explained below.

Many COTS products require adaptation and tailoring for the special requirements of a system and to integrate the product with the rest of the system. Once the product is tailored, it can be distributed to the end users. Licensing of COTS software products can apply to two different versions of the product: the development version which can be adapted using tools associated with the product, and a much less expensive version, called the run-time version, which is distributed to users who will not be able to make modifications.

The basis for cost may be either potential usage or actual use of the software. In the first case, the cost may be determined by the number of CPUs on which the software will be installed or the total number of unique users. Each license may even be tied to a specific CPU and not transferable to a replacement or upgrade, or tied to a specific user by name rather than a "seat." In other cases, the cost basis may be some measure of actual use, such as the number of simultaneous users. In this case, the program management may have to provide "software metering" within the system to monitor the number of concurrent users or to enforce the limits that may be placed on the maximum number of users allowed by the licensing agreement. There are automated tools to help collect data as well as to control the usage of software. These may have to be embedded in and run with the system's software. With the advent of mobile computing, new conditions are being defined for licensing costs. Judicious choice of licenses,

based on the type, number of copies, and frequency of use of software, can assure economical, smooth operation and availability of software.

The services for responding to problems and correcting errors can be included in the conditions of a license. If on-site support is required, then the license may specify locations of these sites. Not all contractors or vendors can support global operations and not all licenses cover global installations of software or even mobile locations. This will directly affect the decisions about organizations responsible for maintenance and the cost of maintenance. Service may come in levels, such as number of calls per month and speed of response to calls to resolve problems. The required availability of a system for operation will affect the decision on what level of service to buy.

Since the licensing area is changing rapidly, it is important for a program to consider all the options, pick the best it can, and allow for renegotiation.

Along with decisions about purchasing and licensing software, programs need to decide when to buy the COTS products. Waiting until the last minute for a large purchase has the advantage of getting the latest version of COTS hardware and software, provided funding can be arranged on that basis. It may have the disadvantage of being a more recent version of the software than the version on which the software was developed, or the development environment itself. Buying the software early may cause faster obsolescence. Buying the software incrementally as each new installation is fielded can create a compatibility problem if the latest versions of software change before all installations are done, and it is no longer possible to buy the earlier compatible versions of either the hardware or software. One strategy is to license a small number of development copies, where appropriate, and wait to license the run-time copies until closer to delivery time.

3.4.4 Organizing and Assigning Responsibilities for Software Maintenance

The plan needs to identify the organizations that will be responsible for performing the COTS-related maintenance activities and their relationship to the organizations performing other software maintenance activities. Either as a part of the purchasing and licensing arrangements, or as a separate arrangement, some agreement will probably be needed with the developer or another organization that can access the source code for a COTS software product. They will help determine if the software product is in error when there is a system problem and, if so, make necessary corrections.

Each program must decide on the best organization and distribution of responsibilities based on the choices available. In some cases, especially with the spiral model of system acquisition, new development and maintenance can occur concurrently. This means that the same organizations may be responsible for development and maintenance. Following are some of the criteria that influence the choice:

- Can economics of scale be achieved by using one organization to maintain more than one system?
- Do security issues limit the choice of organizations to maintain the COTS software? If so, can military personnel be trained to perform maintenance?
- Does the system have high availability requirements that could necessitate on-site support of the COTS products? If so, are the sites in environments that require military personnel or can a contractor provide suitable personnel?
- Is the system deployed on short notice to unpredictable places? If so, some level of organic support may be needed.

The list of COTS-related maintenance activities for which some organization must be responsible can be found in Section 2.3.2. Choices of organizations include organic support dedicated to the program or across programs, an Integrated Product Team, Contractor Logistics Support, vendor support for a specific product, or third-party support for multiple products.

3.5 COTS-Based Maintenance Procedures

The following guidance is for each of the COTS-related procedures performed during the maintenance of a COTS-based system. A brief description of each was given in Section 2.3.2.1.

3.5.1 Market Research

Market research during the maintenance phase of a system involves research to gather information to anticipate changes in the status of COTS hardware and software products already in the system and to find equivalent or better products that could replace those in the system. Market research has to be a continuous process because of the rate at which products change and new products enter the market. The kinds of information that should be gathered from market research include what is shown in Table 3-2. This information can be integrated with the data for COTS product management shown in Section 4. It should again be noted that market research on hardware products must note the impact that changes would have on versions of software in the system, and vice versa.

Table 3-2. Market Research Information

Product Name
Vendor
Version
Expected Release date
New product or update?

Date when support will be discontinued for version in the system (either because of update or product discontinued, or vendor out of business)
Differences between it and version in system (e.g., provides fixes, changes user interface, runs with new version of hardware)
Cost of new or updated product
Date of this report

3.5.2 Upgrade Impact Analysis

Upgrade impact analysis involves looking at the consequences of upgrading or replacing a COTS software product. It also involves analyzing the impact of *not* upgrading. Most of the impacts mean additional work to accommodate the change. Some impacts are counter-productive in the sense that there is no perceived benefit to the program by dealing with those impacts other than maintaining currency of COTS software versions. The impact analysis must weigh the potential benefits against the costs in time, effort, risks, and expense to implement the change. Table 3-3 lists some potential impacts that are direct or side-effects. These were also enumerated under Risks in Section 2.3.1.1

The risk in not upgrading is loss of support from vendors for a product, and the difficulty that may arise in the future if an upgrade must be made that involves moving over several versions instead of just one. This may be a more costly change to make instead of a version-by-version upgrade. Some vendors require payment for all intermediate upgrades when an upgrade is made that skips over several versions. Sometimes you have to keep pace with current versions or you may not be able to migrate the database associated with the COTS product, since vendors typically only provide conversions for one upgrade to the next.

Table 3-3. Potential Areas of Impact due to COTS Product Replacements

Other COTS software products
Data
Adaptation, integration code
Computer hardware
User interface and training of operators
System resource utilization (time, memory)
Software development environment
Software maintenance process
Interoperability with other systems
Operational capabilities

These impacts have to be assessed for each contemplated COTS product replacement and weighed against the benefits. The market research provides some of the inputs for this assessment. Often the impacts listed in Table 3-3 have to be assessed by actually trying the product in a development environment where its effect on the system can be empirically determined. Each of these is discussed briefly below.

3.5.2.1 Other COTS Software Products

Conflicts can occur between a new or updated COTS product and another product in the operational system, especially if they are from different vendors, that cause them to behave improperly or suspend the operation of the system. There can also be an incompatibility between a new version of the operating system and COTS products in the development environment or in the operational system. For example, a compiler that has been used to generate the custom code may not be compatible with a new version of the operating system. This may require updating the compiler or even replacing the compiler with a different one. This, in turn, affects the maintenance process, since all custom software may have to be recompiled. This could result in changes in the size and performance of that software.

Another possible impact of a new COTS software product on the system is its increased functionality that might allow the reduction of custom code or the elimination of some other COTS product. Redesign of a portion of the system may be advantageous in this case.

3.5.2.2 Data

A new version of a COTS software product may change the representation of data it generates or expects to receive. The data may be in a database which must be reconfigured or translated to a new format. The data may be in parameters or messages that pass between products or file formats. The problem can be acute when an operational database already exists although some COTS vendors of database applications may provide conversion aids. Even on the desktop, there have been changes from one version of a word processor or briefing package such as those in Microsoft Office, which have led to incompatibilities in exchanging files between those who have the upgrade and those who don't.

If the size of the files or database are considerably larger, this may force a memory upgrade for a computer or even a computer replacement.

3.5.2.3 Adaptation, Integration Code

Often a substantial amount of code or other specific instructions such as tables, scripts, macros, query language codes, are required to adapt a general purpose COTS product to the specific requirements of a system and to integrate it. If a COTS product changes, this code may also need to be changed. In many ways, it is equivalent to reprogramming custom code but in many different, not necessarily standard, languages.

3.5.2.4 Computer Hardware

If performance and storage requirements of new COTS software installments, and the attendant changes to the custom code, will exceed the capacity of the installed hardware, then it may necessitate improvements to the hardware that have to be coordinated with whoever manages hardware. For some systems this may mean the owners who are part of other organizations who use the system being changed.

It should also be noted that a change in hardware may force a change in versions of software because the current version will not run on the new hardware. Not all vendors move to the new hardware at the same rate so this hardware change may cause some COTS software to be incapacitated for a period of time if the hardware upgrade is not well planned.

When the software is in fact firmware, increases in performance and storage may critically impact the processor in which the firmware is embedded. Practically speaking, the upgrade of the firmware and processor are probably integrated.

3.5.2.5 User Interface and Training of Operators

Some changes to COTS software products can directly affect the user interface. If the operator will have to perform his work in a different way, this may require additional training, including on-line help.

3.5.2.6 System Resource Utilization

In addition to an impact on the CPU and memory in which the hardware executes, a change in the timing for the execution of the software in an upgraded COTS package may cause the

system to fail to meet performance requirements. This, in turn, may require redesign of the system to compensate for the changes.

3.5.2.7 Software Development Environment

The COTS software products in the development environment include those that generate some of the code in the system, e.g., compilers, GUI builders and other application development tools. A change in the version of the compiler may require the recompilation of all of the custom code in the system, re-integration, and re-testing. When these change, the effect can be the same as replacing a COTS product in the operational system and causing any of the other impacts in this list. Other changes in the development environment affect the software maintenance process, described below.

3.5.2.8 Software Maintenance Process

COTS software products to support the software maintenance process include configuration management tools, document generation tools, requirement management tools, and problem reporting packages. Upgrades, or the addition of some of these tools may improve the software maintenance process and make it more efficient. The quality of the documentation with a version of the COTS product will also impact the software maintenance process.

3.5.2.9 Interoperability with Other Systems

Any COTS software changes that affect the interface of a system with other systems or a requirement to have processing that produces the same results as another system will impact the interoperability among these systems unless the changes are synchronized.

3.5.2.10 Operational Capabilities

COTS software at any level in the system may be instrumental in providing operational capabilities. A COTS product upgrade may eliminate some capability that had been used in the system as well as modify a capability that the system depends on.

3.5.3 Asset Management

Asset management for COTS software is concerned with the financial aspects of using COTS software. Among the services that asset management provides is an inventory of what products are being used, where they are located, any other information that is needed to support the management of licenses associated with the COTS software. This is an ongoing activity that involves tracking conformance to the conditions of the licenses including software metering or administering permissions to use packages. Other responsibilities are managing license renewals by alerting those who have to renew them, and monitoring changes in licensing options that may become available so that existing licensing agreements can be modified for financial savings. It is especially important to carefully manage license renewals because some software comes with built-in termination capability when a license expires. This has been known to stop the operation of a military system when the expiration date was reached without a renewal.

3.5.4 Program Planning and Budgeting

As was pointed out earlier, in Section 2.3.1.2, the spending profile for COTS-based systems is different because of the need to be responsive to changes in the commercial marketplace. In order to meet the DOD's planning cycle for funds, the program management must try to plan the cost of maintenance of COTS-based systems so that it is predictable into the future. Elements of software cost associated with the maintenance of COTS software in COTS-based systems include:

- License renewals (roughly 15% of original cost annually)
- Upgrading changes due to impacts
- Retesting
- Retraining
- Market research
- Impact analysis and assessments
- Asset management

One basis for estimating steady state costs will be the information on license renewals that should be in the asset management system. There are several strategies that have been suggested for controlling costs for software upgrades in COTS-based business systems that may be applicable to military systems depending on their specific characteristics. Following are some helpful strategies:

- Plan for a regular update cycle for COTS software every two or three years, with no updates in-between unless it is absolutely necessary. This may lead to a difficult and expensive upgrade if several versions of software have been missed in the interim. It can also cause a re-examination of the COTS products in the system and evaluation of a new design with better products. This approach keeps maintenance costs lower between upgrades.
- Minimize the number of different vendors whose products are used in the system to minimize the number of relationships that have to be maintained with vendors. If there is an integrated set of products, the upgrades will probably maintain their compatibility. This will eliminate some of the integration cost and can provide a single price for a set of products.
- Synchronize updates of COTS software with system releases and estimate the portion of software that will have to be replaced based on expected rate of upgrade of products in the system.
- Plan for computer hardware replacements on a regular cycle such as every three to four years both because of obsolescence and because of the propensity of software vendors to upgrade the software so it no longer runs on older hardware.

3.5.5 Quality Control

Quality control involves maintaining the correct behavior of the system and keeping it available. We have already noted that it is often essential to maintain a relationship with the vendor, usually through the license, to determine when problems are due to the COTS software product or the code associated with its use. The maintenance service may enable the program to submit trouble reports and wait for responses and possibly patches. Some programs hire consultants who are experts in the use of a product to help plan upgrades and to help with problems. These experts have greater credibility with the vendor so they can accelerate acknowledgment of problems by the vendor. For a large or critical system and product, it may be advantageous to have an expert as part of the maintenance team. If the product is used and maintained in many places, it may also be advantageous to have a Help Desk service for the maintainers to consult. For some complex products with a large number of commercial users, there may be a user's group. The program can have representation in the group to help influence the vendor to correct deficiencies and improve the product.

One of the often-cited advantages of COTS products is the large number of users who help to find errors so that the quality of the product improves. This does lead to a phenomenon that the maintenance organization must contend with; namely, that the vendor will send out a stream of patches to the software to fix errors found by other users. The maintainer must decide which to incorporate, and when. The fixes may be in areas of the COTS package that are not used by the system, or could potentially lead to other errors because of the way that the system uses the COTS package. The maintainers must find a low-risk way of dealing with these patches.

One of the difficult problems in maintaining the quality of the system when a COTS software product is replaced is how much to test. Clearly, only black box testing can be done. When the product is replaced, it is important to test for any side effects that may not be directly at the interface of the product with the rest of the system.

Collecting problem reports and analyzing them is an important part of quality control for COTS software. The data should show when the problem was due to an error in a COTS product and how long it took for a correction. This kind of information can show trends in the reliability of each product and the kind of service the maintainer of that product is giving. These metrics are useful in determining when a product needs to be replaced because of its poor performance or that of its vendor.

3.5.6 Configuration Management

Configuration management is important for any system. In the case of COTS components, the configuration management must be down to the level of each COTS software product, and each platform on which a copy is installed or used through a client/server relationship. The inventory for asset management may be adequate for this purpose. Section 4 of this report gives information that can be used for configuration management of COTS software.

This information is useful in distributing software fixes or upgrades to multiple sites, or restoring software configurations that have been destroyed. If patches are frequently received from the vendor and added to the COTS products, they must be recorded in the configuration management system as well.

Because it is easy for operators at remote sites to obtain their own copies of some COTS products, particularly those that are shrink-wrapped and don't involve licenses, the configuration management activity must be sure that these operators do not introduce local changes to the configurations without formal configuration management.

3.5.7 Logistics Support

As stated earlier, logistics support is defined here to include the acquisition of copies of upgraded COTS software products, their distribution and installation, and re-installation if necessary. Sometimes only one copy of an upgrade is received from the vendor, and the user must make the copies and distribute them. The distribution of software to remote sites that are networked can be automated and centrally executed in many cases, unlike hardware that must be stocked and shipped. There are software distribution systems already on the market that will uninstall prior versions of software and install new versions at remote sites over networks, even when the configurations are not identical.

Section 4

Managing COTS Product Information

The key to managing the maintenance of a COTS-based system is to establish and maintain a set of information about the COTS products in the system and to use that information as the basis for decisions and actions. Table 4-1 is a list of the information that should be tracked for each COTS product or version in the system.

Table 4-1. COTS Product Database

Identification	Product name and version number
Vendor information	How to contact vendor or supplier
Platform	Hardware configuration
Options	Any specialized options
Asset management information	License conditions (e.g., services, warranties, organization providing service)
-- License data	License expiration date
	Organization maintaining license
-- Financial data	Financial obligation (e.g., cost of current license and cost basis)
	Number of copies
-- Support status	Support Status: (e.g., version no longer supported, not latest version, latest version, date when support expected to end)
	Expected date of next release or vendor no longer supports the product
Configuration information	Where does the product reside (which specific platforms)
Reliability information	Problem report and error information attributed to this product (e.g., number of errors found over time, severity, time required to fix errors, number of outstanding problems or errors)
System integration information	Role of product in system (e.g., development tool, operational infrastructure, application)
-- Criticality to operation or maintenance	Criticality of COTS product to system operation or maintenance (e.g., critical to operation of system or its interoperability, provides key application functions, affects operator interface etc.)
-- Dependency, technical impact information	Degree of integration with other parts of system (e.g., part of the basic infrastructure, has interfaces with a high number of system components, separable)
-- Amount of adaptation, integration software	Amount of software that is written to adapt a COTS package to system configuration, functionality, and interface with other components
-- Conformance to standards	Use of standard API, modified standard, proprietary standard

Bibliography

The following are documents that supplement the information in the report. They are all on the Internet. Their URLs are given.

- Defense Information Systems Agency (DISA) Defense Information Infrastructure (DII) Common Operating Environment (COE) [http://spider.osfl.disa.mil/dii/]

  This Web site contains links to information on the DII Common Operating Environment as well as systems that use its software, such as Global Command and Control System (GCCS) and Global Combat Support System (GCSS).

- COTS Inclusion in the DII COE", DISA, undated [http://coeeng.ncr.disa.mil/REFERENCE_PAGES/JCSCOT/JCSCOT.HTM]

  This paper discusses considerations surrounding the inclusion of commercial, off-the-shelf (COTS) products in the Defense Information Infrastructure (DII) Common Operating Environment (COE).

- A Risk-Mitigating Model For Large-Scale COTS (Commercial-Off-The-Shelf) Integrated Software Systems, Vu Tran, Thinking Systems, Inc., June 1998 [http://www.ts.com/papers/developi.htm]

  The COTS-based Integrated System Development (CISD) model addresses the needs for a more integration-centric approach to software development.

- COCOTS (COnstructive COTS) Integration Cost Model: An Overview, University of Southern California Center for Software Engineering, March 1998 [http://sunset.usc.edu/COCOTS/docs/cocotsARR030998/index.htm]

  This presentation describes COTS Integration Cost Sources, COCOTS vs. COCOMO Cost Sources, COTS Assessment, COTS Tailoring, COTS Glue Code Development and Test, COTS Volatility Effects on Application, Development Cost, Total COTS Integration Cost.

- SEL Package-Based System Development Process, Software Engineering Lab, February 1996 [http://fdd.gsfc.nasa.gov/cotsweb.pdf]

  The Software Engineering Lab at NASA Goddard Space Flight Center (SEL) Package-Based System Development Process describes differences between the new integration-based life cycle and a traditional development-based life cycle, and recommends best practices for developing COTS-based systems and for managing software projects in the changing environment.

Distribution List

Internal

D200
J. F. Hill

D300
R. W. Bond

D310
D. E. MacLure
G. R. Norris (10)

D340
G. S. Borrelli
H. E. Arbaczawski

D370
S. H. Farish
G. E. Smith

D380
M. J. Bloom
L. M. Rosa
A. M. Willhite

D440
R. O. Pettibone

D470
J. L. Higginson

D500
S. R. Ames, Jr.
T. K. Backman
H. A. Bayard
J. A. Clapp (5)

D510
C. D. Bowen
D. L. Cuomo
M. J. Hebert
D. O. Norman
J. A. Wilson

D520
R. A. Modeen

D530
J. R. Knobel
C. Loizides
R. A. Martin

D540
R. A. Cherinka
J. G. Scarano
M. J. Heller

D550
J. A. Maurer
T. M. Wheeler

D560
T. R. Aiken
H. C. Masterman

D570
C. L. Peterson
A. Sateriale
E. C.Wigfield

D580
D. A. Drake
R. W. Noel

D630
M. Miana

D640
P. B. Hennessey
R. M. Weigand

D710
D. C. Miller
J. L. Monti

L. R. Novak

G04F
J. S. Wood

G04J
S. D. Wallace

R108
D. F. Arimento

W062
S. Gross

W063
R. A. Francis
A. E. Taub (5)