

Guide

Managing the Use of Commercial Off the Shelf (COTS) Software Components for Mission-Critical Systems

AIAA standards are copyrighted by the American Institute of Aeronautics and Astronautics (AIAA), 1801 Alexander Bell Drive, Reston, VA 20191-4344 USA. All rights reserved.

AIAA grants you a license as follows: The right to download an electronic file of this AIAA standard for temporary storage on one computer for purposes of viewing, and/or printing one copy of the AIAA standard for individual use. Neither the electronic file nor the hard copy print may be reproduced in any way. In addition, the electronic file may not be distributed elsewhere over computer networks or otherwise. The hard copy print may only be distributed to other employees for their internal use within your organization.

Guide

Managing the Use of Commercial Off the Shelf (COTS) Software Components for Mission-Critical Systems

Sponsored by

American Institute of Aeronautics and Astronautics

Approved

13 October 2006

Abstract

The purpose of this Guide is to assist development and maintenance projects (teams and individuals) that have to address the use of, or consideration of, COTS products within large, complex systems, including but not limited to mission critical systems. This assistance is provided by capturing a set of information about COTS products (benefits, risks, recommended practices, lifecycle activity impacts) and mission critical systems (variety of MCS, special needs for MCS, differences between MCS and other types of systems) and then providing some linkage between these topics so that various types of stakeholders can find useful information. The document should be of value to both management and technical individuals/teams. It should also be of value to teams that are dealing with non-MCS, in that the scope is not limited to only MCS.

Managing the use of commercial off the shelf (COTS) software components for mission-critical systems :
AIAA G-118-2006 guide.

p. cm.

Includes bibliographical references.

ISBN 1-56347-915-X (print) -- ISBN 1-56347-916-8 (electronic)

1. Aeronautics--Computer programs. 2. Astronautics--Computer programs. 3. Astronautics--United States--Equipment and supplies. 4. Software engineering. 5. Commercial products. I. American Institute of Aeronautics and Astronautics.

TL563.M36 2006

629.10285'5--dc22

2006039308

Published by
American Institute of Aeronautics and Astronautics
1801 Alexander Bell Drive, Reston, VA 20191

Copyright © 2006 American Institute of Aeronautics and
Astronautics
All rights reserved

No part of this publication may be reproduced in any form, in an electronic retrieval system
or otherwise, without prior written permission of the publisher.

Printed in the United States of America

Contents

Foreword.....	vi
Introduction	viii
1 Scope	1
2 Applicable Documents	1
3 Vocabulary.....	1
3.1 Acronyms and Abbreviated Terms.....	1
3.2 Terms and Definitions	2
4 Document Roadmap	2
5 The COTS Decision: Background and Overview	4
5.1 Why COTS?.....	4
5.2 Mission Critical Systems and Requirements	5
5.2.1 Characteristics of Mission Critical Systems (MCS)	5
5.2.2 Mission Critical System Requirements.....	6
6 COTS Software Benefits and Risks	7
6.1 COTS Benefits.....	7
6.2 COTS Risks	8
6.2.1 Incorrect 'Make Versus Buy' Trade Studies.....	8
6.2.2 System incompatibilities.....	8
6.2.3 Human/Machine Interfaces and Operational Procedures.....	8
6.2.4 Underestimated Total Program Costs.....	9
6.2.5 Impacts to Lifecycle Model Activities.....	9
6.2.6 Product Volatility.....	9
6.2.7 Vendor Viability.....	9
6.2.8 Product Support.....	9
6.2.9 Likelihood of 'Wrappers'.....	9
6.2.10 Product Testing/Evaluation	10
6.2.11 Multiple COTS Product Integration.....	10
6.2.12 Problem Resolution.....	10
6.2.13 Product Knowledge/Training.....	10
6.2.14 Product Version Control/CM	10
6.2.15 Product and Process Insights.....	10
6.2.16 Dormant Code	11
7 Recommended Practices.....	11
7.1 Effective Trade Studies	13
7.1.1 The 'Make Versus Buy' Decision.....	13

7.1.2	Product Selection/Evaluation	13
7.2	Insights.....	14
7.2.1	Insights Into the COTS Products	14
7.2.2	Insights Into the COTS Vendor's Development/Maintenance Processes.....	14
7.2.3	Insight Into the COTS Vendor's Business Viability.....	14
7.2.4	Insight Into the COTS Marketplace	15
7.3	Early and Ongoing Testing and Prototyping	15
7.4	Assessing Impacts of "Wrapper" Software	16
7.5	Problem Resolution	16
7.6	Product Knowledge Acquisition	16
7.7	Product Support.....	17
7.8	Configuration Management and Version Control	17
7.9	Cost Estimation Effectiveness	17
8	COTS Activities Throughout the Life Cycle.....	17
8.1	Mapping COTS Software Activities to the Primary Life Cycle Development Process.....	18
8.2	The COTS Software Life Cycle.....	19
8.2.1	COTS Life Cycle Partitions	19
8.2.2	COTS Software Activities.....	19
8.3	Mapping Risks to COTS Software Activities	20
8.4	COTS Specific Activities within each Partition of the Software Life Cycle	24
8.4.1	Product Feasibility	24
8.4.2	Product Evaluation and Selection.....	26
8.4.3	Testing, Integration, Acquisition and Maintenance.....	27
9	COTS Software Activities in the Context of Software Life Cycle Processes.....	28
9.1	Summary of Process Activity Modifications	28
9.2	Primary Life Cycle Process Modifications.....	32
9.2.1	Acquisition Process Modifications	32
9.2.2	Supply Process Modifications (Contract)	33
9.2.3	Development Process Modifications	33
9.2.4	Operation Process Modifications.....	48
9.2.5	Maintenance Process Modifications.....	48
9.3	Supporting Process Modifications	48
9.3.1	Documentation	48
9.3.2	CM.....	49
9.3.3	Problem resolution	50
10	Examples, Case Studies.....	50

10.1	COTS Certification Plan Example	50
10.2	COTS in Enterprise Wide Financial Management System	50
11	Summary and conclusions.....	51
12	Bibliography	51

Figures

Figure 1	— G-118 Roadmap	3
Figure 2	— Synchronization of COTS Software Activities and the Software Life Cycle Development Process	18
Figure 3	— Example Custom-Built Software-Centric Waterfall Life Cycle Model.....	34
Figure 4	— Example COTS Software-Centric Waterfall Life Cycle Model.....	35

Tables

Table 1	— G-118 User Perspectives	4
Table 2	— Mission Critical versus Non-Mission Critical System Characteristics	5
Table 3	— Criticality Level Mappings.....	6
Table 4	— Applicable Mitigation Strategies for Risks	11
Table 5	— Life Cycle Partitions	19
Table 6	— COTS Software Activities	20
Table 7	— COTS Activity Related Risks for Product Feasibility Partition	21
Table 8	— COTS Activity Related Risks for Product Evaluation and Selection Partition.....	22
Table 9	— COTS Activity Related Risks for Testing and Maintenance Partition.....	23
Table 9	— COTS Activity Related Risks for Testing and Maintenance Partition (continued).....	24
Table 10	— Summary of Acquisition Process Modifications for COTS.....	29
Table 11	— Summary of Supply Process Modifications for COTS	29
Table 12	— Summary of Development Process Modifications for COTS.....	30
Table 13	— Summary of Operation Process Modifications for COTS	31
Table 14	— Summary of Maintenance Process Modifications for COTS	31
Table 15	— Summary of CM Process Modifications for COTS.....	32
Table 16	— Summary of Problem resolution Process Modifications for COTS	32
Table 17	— Summary of Training Process Modifications for COTS	32
Table 18	— Examples of COTS Effects on System Requirements.....	37
Table 19	— Examples of COTS Effects on Software Requirements	41
Table 20	— Potential Software Requirements for Particular Types of COTS Software.....	43

Foreword

Commercial off-the-shelf (COTS) products are being considered for inclusion in ever more complex and critical systems. There are clear advantages to considering the use of COTS in such complex or critical systems but given the rigorous needs of such critical systems or subsystems, there have begun to emerge concerns about the suitability of COTS for such applications. This guide identifies some of the characteristics of mission critical systems (MCS) that make the selection process of COTS products (hardware, software, subsystems, etc., but especially software) an emerging success factor in total system acceptance.

Mission critical system characteristics such as reliability, safety, availability, maintainability and certification tend to influence whether or not COTS should be considered for a given application. Once the suitability of COTS has been determined it may be possible to place additional requirements on the product and/or the product's vendor for such mission critical applications. Further, it is possible that certain system requirements and expectations may need to be modified because of the inclusion of COTS products into that system. As COTS products continue to be considered as candidates for inclusion within MCS, there are likely to be additional concerns and factors to emerge that will influence how both buyers and vendors decide if and/or when to use COTS products. The benefits and risks of using COTS in MCS are detailed in this guide. Risk mitigation approaches applicable to the selection and usage of COTS software components in MCS are discussed as they apply to the software product lifecycle.

This guide was initially focused on the use of COTS software products in MCS. As it was developed, it quickly became clear that some/much of what was being captured applied to COTS products other than just software. It also became evident to the committee that much of what was being captured applied to non-MCS. This guide will be most useful and applicable to large, complex systems that are considering or actually using COTS software products. The user is also likely to find this guide of value in COTS hardware projects as well as to systems that range from spacecraft and aerospace applications to human resource or financial applications. In addition, this guide can apply to test hardware and software.

This Guide was produced under the auspices of the Software Systems Technical Committee (SSTC) of the American Institute of Aeronautics and Astronautics (AIAA). It was initially conceived by Ronald J. Kohl, of R. J. Kohl & Associates who provided substantial content. The document was then transferred to the AIAA Software Systems Committee on Standards (CoS) for further review and formal approval as an AIAA standards document.

At the time of approval, the members of the AIAA **Software Systems CoS** were:

Ronald J. Kohl, Chair	R.J. Kohl & Associates
Rick Kwan	Aerospace Computing, Inc.
Joe Marshall	BAE Systems
A. Terry Morris	NASA Langley Research Center
Shawn Rahmani	The Boeing Company
Nancy Sodano	The Charles Stark Draper Laboratory, Inc.

The above consensus body approved this document in October 2006.

The AIAA Standards Executive Council (Amr ElSawy, Chairman) accepted the document for publication in October 2006.

The AIAA Standards Procedures dictates that all approved Standards, Recommended Practices, and Guides are advisory only. Their use by anyone engaged in industry or trade is entirely voluntary. There is no agreement to adhere to any AIAA standards publication and no commitment to conform to or be guided by standards reports. In formulating, revising, and approving standards publications, the

committees on standards will not consider patents that may apply to the subject matter. Prospective users of the publications are responsible for protecting themselves against liability for infringement of patents or copyright or both.

Introduction

It should be noted that although the intent of this Guide is to focus on COTS software products and their use in large, complex software-based systems, it does contain guidance that can apply to certain aspects of COTS hardware utilization. Some of this guidance can also be applied to non-developed items (NDI) types of software other than COTS, such as open source software, reusable component software and even shareware/freeware software. These other applications are not highlighted or discussed specifically herein but are mentioned to encourage users to think broadly and across many problematic areas to determine if there is additional value contained in this document that may be applicable.

It should be noted that this guide may be used for managing the risks associated with the use of COTS products in large, complex systems utilizing the mapping of the 'Recommended Practices' to the 'Risk areas'. Sections 8 and 9 discuss how these various practices and processes can be applied across the software systems lifecycle, therefore helping to establish or improve the set of software and systems development processes.

1 Scope

The purpose of this Guide is to assist development and maintenance projects (teams and individuals) that have to address the use of, or consideration of, COTS products within large, complex systems, including but not limited to mission critical systems. This assistance is provided by capturing a set of information about COTS products (benefits, risks, recommended practices, lifecycle activity impacts) and mission critical systems (variety of MCS, special needs for MCS, differences between MCS and other types of systems) and then providing some linkage between these topics so that various types of stakeholders can find useful information. The document should be of value to both management and technical individuals/teams. It should also be of value to teams that are dealing with non-MCS, in that the scope is not limited to only MCS.

2 Applicable Documents

The following documents contain provisions, which, through reference in this text, constitute provisions of this guide. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this guide are encouraged to investigate the possibility of applying the most recent editions of the applicable documents indicated below. For undated references, the latest edition of the applicable document referred to applies.

IEEE/EIA 12207	<i>Industry Implementation of ISO/IEC 12207:1995 — Standard for Information Technology — Software Life Cycle Processes</i>
ISO 9126	<i>Software engineering -- Product quality -- Part 1: Quality model</i>
NASA-STD-8719.13B	<i>Software Safety Standard</i>
NASA-GB-8719.13	<i>NASA Software Safety Guidebook</i>
RTCA DO-178B	<i>Software Considerations in Airborne Systems and Equipment Certification</i>

3 Vocabulary

3.1 Acronyms and Abbreviated Terms

AIAA	American Institute of Aeronautics and Astronautics
CM	Configuration Management
CBS	COTS-based System or Systems
CMM	Capability Maturity Model
COTS	Commercial Off-The-Shelf
DOD	Department of Defense
FAA	Federal Aviation Administration
MCS	Mission Critical System or Systems
MOTS	Modified Off The Shelf
NAD	No Additional Detail provided
NASA	National Aeronautics and Space Administration
NDI	Non Development Items

OS	Operating System
PC	Personal Computer
R&D	Research & Development
TBSL	To Be Supplied Later, in a future revision

3.2 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

Commercial Off-The-Shelf (COTS)

a product that is

- A commercially available product
- Available in multiple identical copies
- Used without modification of internals
- Supported and evolved by a vendor

Mission Critical System (MCS)

any system (or subsystem or component) that is essential to an enterprise's or organization's or a project's overall success

NOTE These systems, subsystems or components could be hardware, software, procedures or people. This guidebook will focus on the software components and products within such MCS.

Tailoring

Changing parameters and other mechanisms the vendor has made available as part of the product

Modifying

Changing something about the product, particularly the source code, to do something different from the vendor's original intent

4 Document Roadmap

Different decisions exist on a given COTS based systems project. These include: whether or not to use COTS software components; selecting which ones to use; and integrating their use effectively with the overall system development, operation, and maintenance processes. This document is organized to provide easy access to information relevant to those decisions.

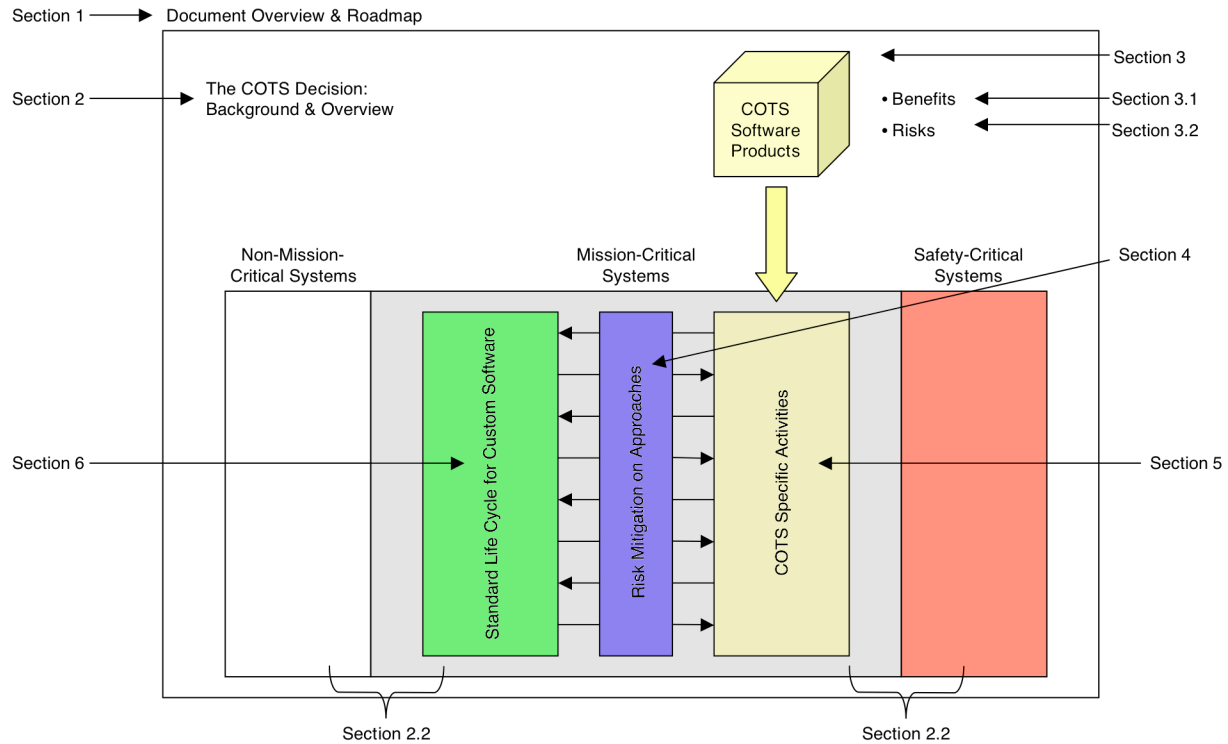


Figure 1 — G-118 Roadmap

This document is organized as follows:

- Sections 1-4 introduce the document, explain its scope, and help readers identify sections of interest.
- Section 5 provides a context for the COTS decisions. It contains a discussion about the differences in custom built software and COTS software products and attempts to identify a few key factors that contribute to the different risks associated with COTS products as well as the reasons for changes in lifecycle activities.
- Section 6 presents an overview of the benefits and risks associated with using COTS software components.
- Section 7 identifies a set of 'Recommended Practices' which are intended to improve the existing set of technical processes for a COTS-based systems project and which can be used to mitigate the risks described in section 6.
- Section 8 defines the major COTS-specific activities needed in the life cycle, such as product evaluation and selection, which do not usually occur for custom-built software.
- Section 9 places the activities in section 8 into the context of a standard (IEEE/EIA 12207) set of software life cycle processes. For each process in the standard it details other minor modifications, which are necessary to incorporate COTS software components.
- Section 10 collects illustrations, examples and case studies of process execution, lessons learned, or other experiences with COTS software components. Contents of this section are contributed by users so that others may benefit from their experiences.
- Section 11 summarizes the document and presents future work planned for later revisions as well as places where user feedback is requested.
- Section 12 contains an extensive bibliography on COTS software-related topics, organized by topic.

In a process standard describing considerations for using COTS software there are various perspectives of the decision that must be incorporated to be meaningful. This guide considers a variety of user perspectives. These include users new to COTS software products, novice integrators, managers, as well as experienced users accustomed to formal lifecycle integration. This guide has been designed to address some of the needs, though not in exhaustive detail, of the different user types. Table 1 provides a general road map for users with respect to this guidebook. Users with specific interest will be directed to “Suggested parts of the Guide” for appropriate usage of its components. The suggested parts in Table 1 that are described in **bold** font are considered to be the primary focus for the particular user. The suggested sections without bold font are considered references that provide supporting information to the primary focus. Readers generally familiar with the use of COTS software components, but desiring more in-depth information on a particular topic may find the detailed bibliography most helpful.

Table 1 — G-118 User Perspectives

Class of User	Perspective and Interests	Suggested parts of Guide
New to COTS Software	Inexperienced with COTS software but curious about the COTS software decision and the pros and cons of COTS software products	5, 6
Novice Integrators	Inexperienced with COTS software but need to integrate COTS software into their systems	5, 6, 7, 8, 10
Managers	Sponsors that need an overall view of the COTS software decision and risks within the general lifecycle	5, 6, 8, 10
Experienced & Formal Integrators	Experienced with COTS software and need to integrate COTS software using a formal lifecycle process	6, 7, 8, 9, 10

5 The COTS Decision: Background and Overview

5.1 Why COTS?

When software systems developers and maintainers start to consider the use of COTS products, either in place of or in addition to custom developed software, the primary rationale is the reduction in schedule, cost and/or risk. It is generally agreed that it is cheaper and faster to buy a product than to build the equivalent of the product from scratch. But just as there are benefits to considering COTS products (see Section 6.1), there are also risks associated with such considerations and selections (see Section 6.2). The challenge to the systems developer/maintainer is to adequately consider the balance between these benefits and risks.

The situation can become more challenging when dealing with the more stringent, unique and inflexible nature of mission critical systems (see Section 5.2). In MCS, it is not unusual for unique and specialized requirements to be established. This makes the use of COTS products even more challenging. Further, there can be requirements on the development environment (compilers, test tools, etc.) for MCS that place additional burden on COTS products that are used in these development environments, as well as in the operational system.

For these reasons it is very important that sound engineering practices be applied to the consideration of, the selection of and the use of COTS products in large, complex systems, especially in mission critical systems.

5.2 Mission Critical Systems and Requirements

5.2.1 Characteristics of Mission Critical Systems (MCS)

Section 3.2 defined a 'Mission Critical System' as any system (or subsystem or component) that is essential to an enterprise's or organization's or a project's overall mission success. These systems, subsystems or components could be hardware, software, procedures or people. This document focuses on the software components and products within such MCSs. Such systems typically have greater demands on them and expectations of them, or other distinguishing characteristics that range from safety, reliability, availability and performance to backup/recovery needs, maintainability and security. The nature of these system attributes tends to vary by application domain but there are certain subsets of these demands that tend to reappear across domains and tend to be consistent over time. The selection of a COTS product that meets requirements at a point in time is important but it must be managed continuously as updates to the products are released and this is particularly important for MCS systems that must maintain consistency across these releases. While it is accepted that any safety critical system is an example of a MCS, there are many other non-safety critical systems that are legitimately considered MCS by the owners/developers/operators of those systems. An example would be a financial management system responsible for very large fund transfers between financial institutions.

See Table 2 for examples of such MCS characteristics and how they might differ between mission critical systems and non-mission critical systems. These MCSs characteristics surface in both the development phase of MCS's as well as the operational phase and even in the termination/disposal phase. Key to these requirements are phrases like "all the time" which are measured in the number of 9's of probability over long periods of performance (i.e. 99.9999% over 10 years).

Table 2 — Mission Critical versus Non-Mission Critical System Characteristics

Non-Mission Critical	Mission Critical
Meet functional requirements	Meet functional requirements, all the time or graceful degradation
Meet performance requirements, often	Meet performance requirements, all the time
Be available, often	Be available, all the time
Work correctly, most of the time	Work correctly, always
Maintainable	Highest quality of maintenance
Crashes tolerated	Graceful degradation mandatory
Ease of recovery	Rigorous recovery/restore requirements
Little/no interest in development artifacts	Development artifact insights (process, plans)
Industry standards, benchmarks	'Certification' requirements
Short development schedule	Long development schedule
Return policies, refunds, buyer beware	Legal protection, liability, insurance, warranties, penalties

As mentioned in Section 5.1, there are typically several criticality levels for various components within a given mission critical system, ranging from safety critical to inconsequential. These criticality levels can significantly impact the architectural design of systems, from individual components to subassemblies and all the way to major subsystems. It is imperative that any project dealing with criticality levels establishes the criteria that separate each of these criticality levels and then institutes practices and processes to ensure that all criteria are satisfied prior to declaring the system “ready for operations”. Criticality level allocation also impacts the structure and dynamics governing system and subsystem interactions.

Table 3 — Criticality Level Mappings

Criticality Level	COTS Guidebook	FAA RTCA DO-178B	NASA NASA-STD-8719.13A NASA-GB-8719.13		
			Hazard Severity	Risk Index	Safety Analysis Class
1	Software that performs critical functions whose single failure could cause a loss of life and/or the entire mission-critical system	Level A	Catastrophic	1,2,3	Full or Moderate
2	Software that performs critical functions whose single failure could cause a loss of a significant function within the critical system	Levels B & C	Critical or Moderate	3,4,5	Moderate or Minimum
3	All others	Levels D & E	Negligible	6,7	None

Table 3 describes the criticality levels used for mission critical systems software in this guide. The three categorical levels of this document have been loosely mapped to other useful guides and standards used in conjunction with NASA programs [NASA-STD-8719.13A and NASA-GB-8719.13] and the FAA’s five criticality levels established in RTCA DO-178B. G-118 mappings are as follows. Criticality level 1 corresponds to DO-178B Level A and NASA’s catastrophic (safety-critical) hazard severity category. Criticality level 2 maps to DO-178B’s Levels B & C and NASA’s mission-critical or moderate hazard severity. In like manner, criticality level 3 conforms to DO-178B’s Levels D & E and NASA’s negligible criticality severity.

It should also be noted that although G-118 focuses on the operational system, there are sometimes processes and products that are part of the development environment that can manifest issues or risks in the operational system. Examples include software language compilers and code generators, whose output becomes part of the operational system. This guide does not dwell on these developmental or environmental topics that could have impacts on the operational system but notes that they exist and should also be considered.

5.2.2 Mission Critical System Requirements

There exists a challenge in the development of requirements when considering the use of COTS products due to the ‘it does what it does’ nature of COTS products. The challenge is how to develop requirements that satisfy various stakeholder needs yet are not too stringent to exclude candidate COTS products. This

challenge is further exacerbated when dealing with mission critical systems and high priority requirements in that there is typically little or no flexibility in anyone's ability to either modify or delete any such high priority requirements. Yet it is the very nature of COTS products that their requirements are essentially cast in stone once you identify a given COTS product as a candidate for integration into the system. This challenge suggests that a different approach to requirements definition and development needs to be considered. This starts by considering COTS products and their capabilities/performance very early in the lifecycle. By doing so, these COTS product capabilities (i.e. the product requirements) can (and should) be considered as one source of 'systems requirements' for consideration by the 'requirements development' processes. Further, those requirements that are captured from various stakeholders may need to undergo some modification in those cases where a viable COTS product candidate has been identified but which does not explicitly satisfy a given requirement. The ability to prioritize requirements and the willingness to consider modifications to requirements (especially high priority requirements) are key approaches to dealing with this challenge. Note also that sometimes, as COTS products are evaluated for consideration for integration into a mission critical system, it is possible that certain product capabilities for which there are no requirements could be identified as 'desirable/needed' which suggests that the early lifecycle activity of evaluating COTS products can actually be a source of system requirements.

6 COTS Software Benefits and Risks

This section summarizes the potential benefits of using COTS software components in mission critical systems and the risks involved. Since many of the risks occur across the whole life cycle of a system, the later process-specific portions (Sections 8 and 9) contain cross-references to this section for the risks relevant to the particular process.

Section 7 discusses activities that can be used to mitigate the risks. Many of the risk mitigation activities apply to more than one risk and are also cross-referenced to the risks discussed in this section in Table 4.

6.1 COTS Benefits

There tends to be less known about the quality attributes (reliability, availability, etc.) of many COTS products. This presents the challenge of satisfying Mission Critical needs and requirements with COTS products. There are many valid business and technical reasons for considering COTS products in large, complex systems, including Mission Critical Systems.

Potential benefits may include the following.

- Reduced development cost/schedule: it is almost always the case that it is cheaper and faster, from a development-phase perspective, to 'buy' a solution than it is to 'build' a solution.
- Reduced maintenance cost: the vendor has employed the necessary staff and skills to update the product and correct any errors, so that the purchaser's costs for maintenance are the 'upgrade' fee.
- Product 'improvements' at no cost: the vendor is driven by market forces and innovation and, therefore, will make 'improvements' to the product at only an 'upgrade fee' to the purchaser.
- 'Proven' product: as a product remains in the marketplace over time, it can be assumed that this product must have 'proven value' to the marketplace. This value can range from highly desirable features to stability and reliability.
- Wide user base to identify and resolve problems: as a product gains and retains market share, the user base increases and helps to identify and resolve problems.
- Wide user base to build shelf life: as a product gains market share, the user base will develop 'shelf-life' (i.e. operationally successful) suggesting improved stability and reliability.

- Available skill base: as a product gains market share, the skill base needed to support that product will also grow. This will be true for the vendor and hopefully third party support suppliers.
- Industry investment in technology base: the vendor has undertaken any necessary R&D and other technology transfer in order to package the underlying technology for a given COTS product. This is a significant value to the customer.
- Commonality with other systems

6.2 COTS Risks

As with any systems development effort and no matter the type of components and products used in that system development effort, there will be risks. This is no different for COTS products. It is important to have a good and complete understanding of these risks, across the entire of the systems development project lifecycle, so that appropriate risk management actions can be taken.

6.2.1 Incorrect 'Make Versus Buy' Trade Studies

There has been a trend towards the use of COTS products in large, complex systems but sometimes the systems developer and/or customer skip the very important early lifecycle step of considering the 'make versus buy' trade study. There are several possible reasons for incorrect trade studies including limited stakeholder involvement, once and done, inadequate level of effort or critical functions not verified with testing. There are still very valid and objective reasons to chose the custom-built software approach but only an effective 'make versus buy' trade study can help to make this critical systems development decision. It should also be recognized that there are sometimes non-technical factors that may influence a 'make versus buy' decision (e.g. legislative, political, etc.).

6.2.2 System incompatibilities

There are a variety of opportunities for a given COTS product to be incompatible with other system elements. These include: hardware platforms, operational environments, legacy systems, external interfaces. A COTS product may be unfit for the operational environment that it will be inserted/integrated into. This is often due to unreasonable expectations placed on a COTS product by someone not fully educated about the product's limitations and constraints. Most frequently affected are overall systems performance and capacity. A given product can work perfectly fine when operated in a small prototype or test/demo environment, but when put into the field, under significant performance or capacity demands can fail to deliver the same results. It is important that all such possibilities for incompatibility be identified and analyzed to determine if further action is required. Notice that this incompatibility risk can arise at various points during the development and maintenance phases due to changes in a currently selected product.

6.2.3 Human/Machine Interfaces and Operational Procedures

It is quite common that the decision to use a COTS product will influence and impact the operational procedures and the 'look and feel' of a new/modified system. This is typically due to the differences between the customer/developer's desired operational profile and the vendor's intended utilization profile. Since the COTS product 'user interface' is typically driven by the marketplace, it is important to identify any concerns by the 'end users' of any product's 'look and feel' (e.g. background color, location of certain data fields on a display, etc) as well as any impact the operational procedures currently planned for such a system. Such differences need to be identified and addressed early, since these operational differences and 'look and feel' concerns can impact end user systems acceptability and, ultimately, systems utilization. This risk area can become even more significant when there are many, many users that could be affected by some change in their daily operational procedures.

6.2.4 Underestimated Total Program Costs

Given the immature state of systems development using COTS products, it is not surprising that accurate cost models are not fully developed. One specific area of common underestimation is the cost of COTS products during the sustaining engineering phase of a system. In addition, cost impacts such as training and license fees may be overlooked or underestimated as a part of the 'total cost of ownership'. Much more accurate estimates are typically made for the development phase of a system, but there remains some amount of uncertainty as to what kinds of cost impacts will exist for the sustaining engineering phase.

6.2.5 Impacts to Lifecycle Model Activities

It has been recognized that certain new/modified activities are required due to COTS product considerations. Further, some of these new/modified as well as existing activities may need to be performed at different phases in the lifecycle as compared to custom-built software development activities. COTS products impact the initial requirements capture activities in that sometimes the COTS product features become the requirements. Clearly there is much less actual software development for COTS-based systems. Systems integration typically is more complex and lengthy due to unexpected interface issues arising in COTS-based systems. COTS product re-evaluations should be performed throughout the development and maintenance phases of a system's existence.

6.2.6 Product Volatility

Given that COTS products that may undergo changes annually (or more often) are now being used in very long-lived systems, it is important to understand the COTS product change cycle. It is likely that a given COTS product will undergo some amount of change (capability, performance, etc) during the development phase and even more so during the sustaining engineering phase. In addition, the vendor may make product changes based on their desire to 'fix a problem'. The user of a COTS product must be certain that a vendor provided 'fix' does not have a detrimental effect on the system under development or operations. Therefore, this volatility must be acknowledged early in the lifecycle and reviewed and re-analyzed periodically. This risk must also be factored into planning for systems evolution and maintenance.

6.2.7 Vendor Viability

Vendors can come and go. Therefore, it is important to try to gain as much insight into a vendor's future business viability as possible. In addition, some vendors are susceptible to takeover, possibly by an off shore enterprise or a competitor. Of equal concern, there is the vendor who gets bought out by its competitor, only to have the COTS product buried. These risks suggest that business insight and vendor alternative knowledge is important to the long-term success of COTS-based systems.

6.2.8 Product Support

Although COTS vendors provide a certain level of product support, it is not uncommon for such products used in MCS to require additional needs for support, both during the system development and especially during systems operations. Such product support requirements need to surface early. Be aware that vendors can drop support for a given version of a product after some amount of time or after a certain number of additional version releases. In addition, product support can be affected if a vendor goes out of business or even if a vendor is 'acquired' by a larger corporation (especially if this is an off-shore company, which creates a potential need to involve the U.S. Department of State).

6.2.9 Likelihood of 'Wrappers'

In spite of the best attempts to have all requirements satisfied by a collection of COTS products, it is very rare that this actually occurs. Therefore, the likelihood of having to develop some amount of custom software that helps to integrate a given COTS product with other system elements is very high. We call any such custom s/w 'wrapper' code and it can include features such as simple data translation, new

functionality necessary for a given COTS product to effectively integrate into a larger software system or deactivation of certain COTS product features. There is growing evidence that the cost of this 'wrapper code' is perhaps two to three times more costly than typical custom-built software (See COCOTS references (refs 47 and 48)).

6.2.10 Product Testing/Evaluation

Given the complexities of integrating one (or more) COTS products with other system elements, testing may need to be expanded and extended across the lifecycle. This testing should become part of the initial product evaluation efforts as well as a part of the various testing planned for the system under development. In particular, it is important to try to include 'operational'-like testing, since one of the larger risk areas is the difference between the vendor's operational model and the actual system operational model.

6.2.11 Multiple COTS Product Integration

This risk area is a subset of the above 'systems incompatibility' risk area, but it tends to create additional project impacts if not fully addressed on its own. Integrating multiple COTS products, especially if multiple vendors are involved, provides several risks. It is not uncommon to find mismatches and inconsistencies amongst COTS products, even though such products are 'advertised' to be compatible. And it is often difficult to fix incompatibility problems, since it may be difficult to get one (or more) of the vendors to take responsibility for any necessary changes. In addition, increased attention to configuration management becomes critical.

6.2.12 Problem Resolution

It is often more difficult to identify, track down and resolve problems in systems with COTS products. This is, in part, due to being able to attribute a specific problem to a given COTS product and then to get the COTS vendor to agree that a given problem is actually within their product. This problem becomes more challenging when multiple COTS vendors are involved.

6.2.13 Product Knowledge/Training

Most COTS products require some amount of training for users, integrators and maintainers to become knowledgeable about that product. Such training is typically offered by the vendor but usually is offered at an extra cost and requires some amount of dedicated time and personnel to take advantage of this product training. For other than the simplest of COTS products, undertaking such training without involvement by the vendor raises the risk that incomplete product knowledge will occur.

6.2.14 Product Version Control/CM

For systems that spend many months/years in the development phase of a lifecycle, it is important for the acquirer/integrator to maintain awareness of and controls over the various COTS product versions from a given vendor that are either included in a given system or are being considered for inclusion. Given the product volatility mentioned above, it is possible that a given systems development effort may have to utilize more than one version of a given COTS product. Multiple versions of a given product introduce variability in product support for each version, which impacts both the development and operations phases.

6.2.15 Product and Process Insights

In many systems development efforts, but especially so in mission critical systems development projects, it is desired or even required (by the customer/acquirer) to have insights into the processes, methods and development artifacts that are used to produce the COTS product as well as insights into the product itself (e.g. access to the source code). This is also required of systems development projects subject to compliance with the Software Capability Maturity Model (SW CMM) or other such CMMs. Such insights are typically not available from COTS product vendors.

6.2.16 Dormant Code

There is the possibility that a given COTS product may contain features or capabilities that are unnecessary, undesired or even unacceptable. These features, if activated, can lead to undesirable or unacceptable behavior.

7 Recommended Practices

This section summarizes activities which can be used to improve the overall set of technical processes and practices as well as useful to mitigate the risks discussed in Section 6. Table 4 provides a mapping that associates the Recommended Practices with the risks to which they apply. The recommended practices in this table are not comprehensive.

Table 4 — Applicable Mitigation Strategies for Risks

Risk	Applicable Mitigation Strategies
Make versus buy trade studies	7.1 Effective Trade Studies 7.5 Problem Resolution 7.6 Product Knowledge Acquisition 7.7 Product Support 7.8 Configuration Management and Version Control 7.9 Cost Estimation Effectiveness
System incompatibilities	7.1 Effective Trade Studies 7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software 7.8 Configuration Management and Version Control
Human/Machine Interfaces and Operational Procedures	7.1 Effective Trade Studies 7.2 Insights 7.4 Assessing Impacts of “Wrapper” Software
Underestimated total program costs	7.1 Effective Trade Studies 7.4 Assessing Impacts of “Wrapper” Software 7.5 Problem Resolution 7.6 Product Knowledge Acquisition 7.7 Product Support 7.8 Configuration Management and Version Control 7.9 Cost Estimation Effectiveness
Different lifecycle model	7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software

Table 4 — Applicable Mitigation Strategies for Risks (continued)

Risk	Applicable Mitigation Strategies
Product volatility	7.1 Effective Trade Studies 7.2 Insights
Vendor viability	7.1 Effective Trade Studies 7.2 Insights 7.7 Product Support
Product support	7.1 Effective Trade Studies 7.2 Insights 7.4 Assessing Impacts of “Wrapper” Software
Need for wrapper software	7.2 Insights 7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software 7.5 Problem Resolution 7.6 Product Knowledge Acquisition 7.7 Product Support 7.8 Configuration Management and Version Control 7.9 Cost Estimation Effectiveness
Inadequate testing	7.3 Early and Ongoing Testing and Prototyping
Multiple COTS product integration	7.1 Effective Trade Studies 7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software
Product problem resolution	7.5 Problem Resolution
Product knowledge/training	7.6 Product Knowledge Acquisition
Product version control/CM	7.8 Configuration Management and Version Control
Product and process insight	7.1 Effective Trade Studies 7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software
Operational environment mismatch	7.2 Insights 7.3 Early and Ongoing Testing and Prototyping 7.4 Assessing Impacts of “Wrapper” Software