

COTS-BASED SYSTEMS

Introduction

The content of these pages was generated from the work of the SEI COTS-Based Systems (CBS) initiative. The focus was to learn, mature, and transition principles, methods, and techniques for creating systems from commercial off-the-shelf (COTS) products. We have since expanded our focus to improving the creation and sustainment of systems from any set of largely off-the-shelf (rather than only commercial) constituents.

However, since use of COTS products continues to grow, we are keeping this information available for those who will find our past work helpful. The SEI CBS Initiative addressed the challenges of assembling systems from pre-existing components and of adapting legacy systems to take advantage of a CBS engineering strategy. COTS-based engineering arrives at a solution through a set of iterative activities that preserve flexibility while simultaneously addressing such issues as business processes, product evaluation and acquisition, system evolution, programmatic and technical factors, and vendor and customer relations. We hope that practitioners, managers, and all those involved in adoption of COTS products for their organizations' systems will find this body of knowledge useful.

The COTS Challenge

The design, development, and maintenance of COTS-based systems and the migration of legacy systems toward CBS practices are complex. New products and technologies constantly enter the marketplace. The vendors of existing products work to differentiate their product from those of competitors. This leads to a marketplace characterized by a vast array of products and product claims, extreme quality and capability differences between products, and many product incompatibilities, even when they purport to adhere to the same standards.

For organizations designing and implementing a COTS-based system, or upgrading a legacy system with COTS components, the current market state presents a number of challenges. COTS products come with their own architectural concepts that may not match those of your system. COTS products also have built-in assumptions about how they will be used, which may not match your end users' way of doing things. A major difficulty is to discover the actual technical capabilities of a product or set of competing products, since there is no objective forum for product evaluation. Once individual products are selected, it is difficult to identify and resolve mismatches between products, and to avoid becoming captive to the products of a single vendor or set of vendors ("vendor lock"). Equally difficult but necessary is the ability to forecast what technologies and products will be relevant over the life of the system.

Thus, in designing and constructing a COTS-based system, or in modifying a legacy system to take advantage of COTS products, an organization must answer a number of questions, including the following:

- Which technologies and products are most appropriate?

- How can product mismatches be rectified in our system?
- How can we engineer system attributes such as reliability, security, and performance in spite of decreasing control over individual system components?
- How do we integrate COTS products with the custom code that continues to provide the core of many systems?
- How do we take advantage of COTS while delivering a system that can evolve over a long life-time?

Introduction to COTS-Based Systems

The SEI's COTS-based systems (CBS) work focuses on improving the technologies and practices used for assembling previously existing components (COTS and other non-developmental items) into large software systems, and migrating existing systems toward CBS approaches.

The CBS approach changes the focus of software engineering from one of traditional system specification and construction to one requiring simultaneous consideration of the *system context* (system characteristics such as requirements, cost, schedule, operating and support environments), capabilities of products in the marketplace, and viable architectures and designs.

The effect of this fundamental change is profound. Not only must engineering activities such as requirements specification change to support simultaneous consideration of system context, architecture and design, and the marketplace, but so must acquisition processes and contracting strategies. For example, integration contractors and commercial product vendors must be treated as partners and rewarded for identifying the best value to be achieved through the use of COTS products.

While some organizations have achieved success by embracing the fundamental change, the development of COTS-based systems continues to involve significant technical risk and promises of lower cost, higher reliability and easier modernization are often unfulfilled.

The SEI was challenged with developing a systematic and predictable discipline for the engineering and management of COTS-based systems in order to assist organizations in realizing the broad CBS potential.

Activity Areas

The SEI CBS work has focused on three primary categories of practices:

Product and Technology Evaluation

Even organizations that have never developed a COTS-based system are aware of the complexity of selecting a COTS product. Not only must they consider the qualities of competing products, but they must also determine whether the technologies on which the products are based are sufficiently mature for general use, and whether these technologies are likely to remain viable over the life of the system.

Similarly, organizations implementing CBS strategies for new or legacy systems must consider not only immediate system requirements but also the unceasing evolution of computing and software technology.

Failure to address either of these concerns results in a flawed system: in the first case, a system that does not meet immediate user expectations, and in the latter case a system that follows technological directions that ultimately dead-end.

In order to ensure that systems meet user requirements, product evaluation practices must be developed. Typically, products are described in terms of interfaces that provide access to functionality. Here, standards may provide a frame of reference for comparing the product to generally accepted capabilities. Various approaches have been developed for evaluating products in terms of their interfaces.

However, to determine the fitness of a product for a given use, consideration must be given to more than just the interfaces the product provides. Aspects of performance, reliability, and flexibility, as well as the **implicit assumptions** made by the product about the operating environment must be considered. For example, while examination of the published interface of a product may suggest that it can interoperate with a second product, interoperation may be limited by each product's assumption that it has primary responsibility for handling incoming events. Much of this sort of information is not addressed by standards and is unavailable from product suppliers. Thus, hands-on evaluation to identify such mismatches (alternately called architectural mismatches by Garlan, Allen, and Ockerbloom, and interface mismatches by Wallnau, Clements, and Zaremski) must be a primary option.

Evaluation of a product can also extend to examination of other factors, such as the COTS product supplier or the process used to create and maintain the product. For example, many organizations now insist on ISO 9000 certification for vendors as an indication that the vendor's product has been produced using well-defined practices and procedures.

However, simply composing a system of quality COTS products will not ensure that the right technologies are selected or that a system remains viable over an extended period of operation. The ebb and flow of technologies and related products in the marketplace necessitate strict discipline in identifying, analyzing, and selecting COTS products that incorporate viable technologies. In order to understand the characteristics of a technology, an organization can use representative products to build demonstrators and provide proof-of-concept for use of the technology in specific system scenarios.

Part of our CBS activity has involved identifying sound practices for evaluating COTS technologies. These practices are presented in our Product Evaluation Tutorial and in the technical report *A Process for COTS Software Product Evaluation*.

In addition, we have evaluated techniques to wrap legacy and COTS products and mediate or bridge the differences and gaps between these products. We have investigated technologies (and related COTS products) such as Web browsers, CORBA, COM, and Enterprise JavaBeans (EJB) to determine the feasibility of using them to address our customers' problems.

Acquisition and Management

Making the change to a CBS approach is akin to making the change from being a developer and producer of systems to being a consumer and integrator instead. Many of the changes this yields involve evaluation of technologies and products, and the design and engineering of systems. However, there are also numerous changes that affect how the development and maintenance of such systems are managed.

Becoming an effective consumer and integrator of COTS products necessitates applying new strategies for licensing products, negotiating data rights, estimating system development and maintenance cost, predicting schedules, managing personnel, and identifying and reducing risks. An effective CBS consumer also requires knowledge of how to build sufficient flexibility into procurement and contract documents to allow a variety of creative solutions while at the same time encouraging bidders to selecting appropriate components and strategies.

Effective COTS consumers also must identify appropriate steps for determining when a system (or system component) is a good candidate for migration toward a CBS approach. Implicit in this statement is the expectation that some systems do not represent good candidates, perhaps due to characteristics of the system, the available technologies, or the COTS marketplace.

Carney and Oberndorf identify some of the issues to be considered in making determinations regarding the suitability of adopting a CBS strategy.

Our strategy for identifying best acquisition and management practices requires continual contact with commercial and federal organizations experienced with aspects of CBS approaches. These best practices are described in the book *Managing Software Acquisition: Open Systems and COTS Products*. The information gathered from experts has also served as fodder for the development of many of our products. We have further codified our knowledge into the technical report *EPIC, the Evolutionary Process for Integrating COTS-Based Systems*.

Design and Software Engineering

The engineering of COTS-based systems continues to involve significant technical risk. A good indicator of the as-yet unresolved difficulties involved in building COTS-based systems is the "glue code" used to integrate components. This code is often ad hoc and brittle, but it is needed to repair mismatched assumptions that are exhibited by the components being integrated. Without this glue code, the components would not be integrable, yet as a consequence of the code, COTS-based systems can be difficult to comprehend, less evolvable than intended, and less reliable than the constituent products.

In order to express a number of important activities involved in design and engineering of COTS-based systems, we have developed a reference model that describes the central artifacts of CBS components in various states. COTS-based systems engineering often begins with products that have many unknown qualities and intends to produce a system that supports flexible reassembly with different components. The key ideas expressed by this reference model are:

- Off-the-shelf components, especially COTS products, must usually be treated as black boxes, or at best very opaque boxes. As a result, many properties of components must be discovered through systematic investigation in order to be qualified.
- Once the relevant properties of a component have been discovered, it is possible to identify which properties exhibited by a component are in conflict with other components, or with a system design. These conflicts, or mismatches, must be repaired through component adaptation.

- Once the mismatches between components have been removed, it is possible to assemble them into systems, and to evolve the system through re-assembly with different components. The reference model assigns a prominent role to software architecture in supporting both assembly and re-assembly.

These engineering techniques are not applied in a vacuum; there must be an underpinning of a strong architectural approach and the selection and application of appropriate technologies based on assessing various tradeoffs. Simply put, a CBS approach does not remove the requirement for sound engineering practice.

We have produced books based on these techniques to aid in the COTS engineering process:

Building Systems from Commercial Components

Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices

This reference model is not a process model; it does not imply a sequence of steps. Indeed, our experience is that architectural patterns may suggest a set of properties that must be discovered by component qualification and that, conversely, component properties discovered from qualification may suggest certain architectural patterns.

COTS Monograph Series

Government policies on the acquisition of software-intensive systems are undergoing a shift in emphasis from custom development toward the use of commercial components. In systems where using commercial components is both possible and feasible, it is no longer acceptable for the government to specify, build, and maintain a large base of comparable proprietary components.

Like any solution to any problem, there are drawbacks as well as benefits: there are significant tradeoffs when using COTS-based solutions. Thus, policies that favor COTS use should be implemented with an understanding of the complex impacts of using commercial products.

In response to this need, the SEI CBS Initiative is preparing a set of monographs that address such issues as

- finding and selecting appropriate commercial products
- identifying decision criteria for migrating to new or emerging COTS technologies
- understanding the ramifications of the CBS approach on system architecture
- developing testing strategies for systems incorporating COTS components

The first monograph in the series (*Assembling Large Systems from COTS Components*) provides an overview of the issues that arise when using COTS. This monograph is intended as a first-level reference for a program manager, who, when faced with a barrage of directives, memos, and policy statements, asks the question, How do I do COTS?

The table below lists the monographs that are currently available in portable document format (PDF).

Monograph Title	Author(s)	Date	Audience	Format
Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities	Carney, D.	6/97	general	PDF
Case Study: Correcting System Failure in a COTS Information System	Hissam, S.	9/97	technical	PDF
Isolating Faults in Complex COTS-Based Systems	Carney, D. and Hissam, S	2/98	technical	PDF
COTS and Open Systems	Oberndorf, P.	2/98	general	PDF
Case Study: Evaluating COTS Products for DoD Information Systems	Sledge, C. and Carney, D.	6/98	management	PDF
Case Study: Significant Schedule Delays in a Complex NDI-Based System	Carney, D.	6/98	general	PDF
A Summary of DOD COTS-Related Policies	Oberndorf, P. and Carney, D	9/98	management	PDF
DoD Security Needs and COTS-Based Systems	Hissam, S.; Carney, D.; Plakosh, D.	9/98	management, technical	PDF

COTS Usage Risk Evaluation

Are you preparing for a project that makes heavy use of commercial software?

Do you need to better understand the potential risks associated with such a program?

The COTS Usage Risk Evaluation (CURE) has been developed to assist organizations in avoiding common mistakes in COTS-based acquisitions. CURE is ideally given during the early stages of a program, when the major key decisions relating to use of COTS products have not yet been made. CURE is a useful technology for any organization that is preparing for a project that is critically dependent on commercial software; it provides insight and understanding into the potential risks associated with such a program.

CURE involves site visits by SEI personnel to the program office and contractor for COTS-based acquisitions. Structured question-and-answer sessions are used to uncover potential risks in the acquisition. Risks are identified, and strategies for mitigating these risks are provided in a final report.

Detailed information about CURE can be found in the SEI technical report *Identifying COTS Product Risks: The COTS Usage Risk Evaluation*.

Who will benefit?

- program managers preparing to start COTS-intensive projects
- contractors preparing to bid on COTS-intensive government contracts
- anyone interested in gaining greater awareness of the risks inherent in COTS-based acquisition

CURE is a focused examination of the COTS-related aspects of a system development project. It is ideally administered during the early stages of a program, even before a specific contractor has been chosen.

CURE is aimed at both the government and the contractor side of a project. It is intended to assist key personnel on both sides in the decision-making and skills that will be required when an acquisition is heavily oriented toward using commercial software.

While the evaluation is aimed at both the government and the contractor, it can be applied individually to any organization that might participate in a COTS-related acquisition. It can also be used by contractors planning to bid on a forthcoming proposal.

Materials

The evaluation is performed through a questionnaire and an on-site visit by an assessment team. The project's personnel complete the questionnaire and return it to the evaluation team in advance. This permits the team to identify key topics, and to focus the on-site visit toward the individual needs of the program.

The CURE process produces a detailed outbrief on the COTS-related risks to the project. In the outbrief, identified risks are explained and prioritized, and mitigation strategies suggested for each. The outbrief is delivered within two weeks of the on-site visit.

Availability

CURE is offered on demand at customer sites. CURE is separately administered to the acquirer and the contractor. Each individual administration requires no more than two days on site.

Prerequisites

The participants in CURE should be the senior organization members who will be assigned to the forthcoming (or ongoing) project. This includes the acquirer's program manager, the contractor's project manager, and the contractor's lead engineer/chief technical architect.

For More Information

If you are interested in learning more about CURE, please contact us at:

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-2612
cure@sei.cmu.edu.

CURE Analysis Example

The analysis component of CURE is least well described in existing documents largely because the CURE developers have always assumed a formal training program would be in place. However, the following outlines the steps of the analysis process.

1. The three members of the evaluation team go through the evaluation record in order and agree on the text for each risk factor. As the text is agreed the equivalent field in the database is edited to match the agreed text. This process, while time consuming, should not generate much discussion other than on what was heard. If the evaluation team agree that a risk factor doesn't apply then it may be deleted from the database.
2. The database is used to generate the report of conditions and risk factors and a copy is printed for each member of the evaluation team.
3. The team then reads the report and agrees whether the condition should be considered a risk, a strength, or is not applicable to the program. It is usually helpful to capture as much of the discussion as possible since it is during this phase that some risk mitigations arise.
4. For each risk the team must specify the bad consequence, the severity of the risk, the supporting evidence (the risk factors), possible mitigations, and the likely owner of the action to mitigate the risk. For each strength the team simply lists the strength and supporting evidence.
5. Develop the outbrief listing all of the above data.

Generally, the analysis process considers conditions that apply only weakly at best. These are generally eliminated. Conditions where every risk factor is positive are considered strengths and the rest of the conditions are considered to be risks.

CURE Components

The overview describes the overall process for the COTS Usage Risk Evaluation (CURE) as seen from the viewpoint of a member of a program to which CURE is applied. The process is supported by the following artifacts:

- Initial questionnaire: a document sent to the program in order that the evaluation team can understand the goals of the program and shape the face-to-face interview.
- Discussion document: a complete list of topics that might be discussed during the interview.
- Evaluation record: a variant of the discussion document that is used by the evaluation team to record the information heard during the interview.
- CURE database: a rudimentary Microsoft Access database (and accompanying image) that supports the evaluation team in the analysis of the data gained from the interview.

CURE Participant Information

Version 3.2

Description

The COTS Usage Risk Evaluation (CURE) assumes that an organization is at some stage of acquiring a COTS-based software system, to be created under contract by another organization. In the most familiar scenario, the former organization would be some part of the federal government and the latter some large industrial contractor. However, there is no assumption that the acquiring organization is necessarily a government agency: the questionnaire is applicable to any acquiring organization. Ideally, both the acquiring and the contracting organizations will participate in separate evaluations, and a program-wide result will be obtained. However, this is not absolutely necessary.

This document is intended for those participating in the CURE. It provides an overview of the three steps of the evaluation. For each step, both the activity and the personnel expected to perform that step are discussed. Finally, it is assumed that the decision to perform CURE has already been made, so no rationale for a program's participation in the CURE is provided. If such rationale is needed, the reader should consult the CURE portion of the SEI website. The evaluation is performed during an onsite visit by the evaluation team.

Initial Questionnaire

Four weeks before the scheduled onsite visit, the evaluation team sends the initial questionnaire to the program's point of contact (PPOC). The completed questionnaire must be returned to the evaluation team no later than one week before the onsite visit.

The questions are general in nature and should require no more than a half day to answer. The answers to the questionnaire are used to inform the evaluation team about the program.

Onsite Visit

About one week before the onsite visit, the evaluation team furnishes the PPOC with the discussion document. This document guides the discussions during the visit and is provided in advance so that program personnel may understand better the nature of the onsite visit.

The normal model of the onsite visit is that the evaluation team meets with the program participants on Monday of the evaluation week. After an initial inbrief by the evaluation team, the discussion document is used to guide the dialog between both teams. The discussion of some topics will perform take longer than others, but each topic is intended to be roughly at the same level of granularity. Sometimes it is desirable to extend the discussion into Tuesday morning, but it is unlikely that the discussion would be any longer.

The onsite interviews are with key personnel from the organization. For a contractor, the interviews are divided between the lead engineer, the project manager, and possibly the organization's contracts officer. For an acquisition organization, the program manager is the main person to be interviewed. Any supporting persons (e.g., the program's contracts officer) may be added as deemed appropriate, but no more than five people should be interviewed.

There is no need for individual interviews, thus it is expected that all personnel will be present for the duration of the onsite visit. Indeed, the joint interview occasionally surfaces differences of opinion between team members; resolving such ambiguities is an important step.

Over the next three days, the evaluation team analyzes the data gathered during the first day(s) for COTS-based risks. The result of the analysis is the basis of the outbrief.

Outbrief

On Friday of the same week as the on-site visit, the evaluation team returns with an outbrief listing the observed COTS-based risks for the program. Each risk is presented with potential consequences and possible mitigations. The outbrief takes one to two hours, depending on the results of the evaluation team's analysis. The outbrief may generate discussions, which are encouraged by the evaluation team—such discussions may illuminate or clarify issues raised during the evaluation.

The outbrief is intended for the personnel that participate in the initial discussions, but the participants should feel free (and are encouraged) to invite other appropriate personnel. The only caveat is that if CURE is to be applied to both the acquirer and the contractor, and that contractor personnel not be present during the acquirer's outbrief.

In all cases, the evaluation team considers the content of the outbrief to be confidential. This restriction only applies to the evaluation team; the program office may distribute the outbrief as circumstances dictate.

Evolutionary Process for Integrating COTS-Based Systems (EPIC)

Does your legacy system provide significant business value?

Are you considering incrementally developing and deploying a modernized system?

Who Will Benefit?

- Organizations that want to avoid the pitfalls of traditional processes in using commercial off-the-shelf (COTS) applications to meet business needs
- Information Technology managers starting legacy system modernization efforts

Description

EPIC is a management and engineering process for building, fielding, and supporting systems that leverage COTS products and other existing components.

COTS products offer the promise of rapid system delivery to end users, shared development costs with other customers, and opportunities for expanding business and mission capabilities. Yet, this promise is often not realized in practice. Many project teams try to use traditional, requirements-driven processes where they define requirements first, and form an architecture—only then do they search for COTS products. Practical experience shows that using COTS products requires new processes—as well as new skills, roles, and responsibilities.

EPIC does not simply evaluate and select the "right" COTS product. EPIC integrates COTS lessons learned and disciplined spiral engineering practice to define, build, field, and support COTS-based solutions that integrate one or more COTS products with legacy software, other pre-existing software components, and any required custom code. EPIC extends the Rational Unified Process® (RUP®) to provide a risk-based spiral framework to accommodate COTS products.

Throughout the life of the system, EPIC links the disparate stakeholders into a coherent team that simultaneously defines and manages tradeoffs among requirements and end-user business processes, system architecture and design, programmatic (i.e., cost and schedule and risk), and capabilities in the commercial marketplace. Throughout a project, EPIC drives from a strategic vision to an implemented and sustained solution by:

- continuously leveraging marketplace forces
- synchronizing system development with any necessary business process changes
- facilitating interaction among stakeholders
- evaluating products and negotiating needs based on hands-on experiments ("try before you buy")
- providing early mitigation of high-priority risks and visibility at key decision points

Materials

EPIC codifies these practices in a structured flow of key activities and artifacts. The detailed report provides

- guides to the goal, objectives, exit criteria, activities, and artifacts for every phase of a project
- guidelines and artifacts that provide pragmatic considerations to guide a number of COTS-unique activities

Availability

Currently, governmental and commercial organizations are piloting the process. The SEI seeks additional organizations for piloting.

To facilitate a project's transition, an EPIC workshop is recommended. Mentoring services in the application of EPIC are also available.

Related Technical Reports

- *Evolutionary Process for Integrating COTS-Based Systems (EPIC): An Overview* (CMU/SEI-2002-TR-009)
- *Evolutionary Process for Integrating COTS-Based Systems (EPIC): Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS)-Based Solutions* (CMU/SEI-2002-TR-005)

For More Information

Contact: Lisa Brownsword
Software Engineering Institute
Carnegie Mellon University
4301 Wilson Blvd, Suite 902 Arlington, VA 22203
Email: llb@sei.cmu.edu

COTS-Based Systems References

Kazman, R. & Port, D. (Eds.), COTS-Based Software Systems, Third International Conference, ICCBSS 2004, Redondo Beach, CA, USA, February 1-4, 2004. Proceedings, *Lecture Notes in Computer Science*, Heidelberg, Germany: Springer-Verlag, 2004. <http://www.informatik.uni-trier.de/~ley/db/conf/iccbss/iccbss2004.html>

Erdogmus, M. & Weng, T. (Eds.), COTS-Based Software Systems, Second International Conference, ICCBSS 2003, Ottawa, Canada, February 10-13, 2003. Proceedings, *Lecture Notes in Computer Science*, Heidelberg, Germany: Springer-Verlag, 2003 <http://www.informatik.uni-trier.de/~ley/db/conf/iccbss/iccbss2003.html>

Dean, J. & Gravel, A. (Eds.), COTS-Based Software Systems: First International Conference, ICCBSS 2002, Orlando, FL, USA, February 4-6, 2002. Proceedings, *Lecture Notes in Computer Science*, Heidelberg, Germany: Springer-Verlag, 2002. <http://www.informatik.uni-trier.de/~ley/db/conf/iccbss/iccbss2002.html>

Garlan, D., Allen, R. and Ockerbloom, J., Architectural Mismatch: or Why It's Hard to Build Systems Out of Existing Parts, *Proceedings of the International Conference on Software Engineering*, Seattle, 1995

Kontio, J., A Case Study in Applying a Systematic Method for COTS Selection," *Proceedings of the International Conference on Software Engineering*, Berlin, 1996

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479

Web: www.sei.cmu.edu | www.cert.org

Email: info@sei.cmu.edu

Copyright 1998-2012 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

COTS Usage Risk EvaluationSM, CURESM, EPICSM and Evolutionary Process for Integrating COTS-Based SystemsSM are service marks of Carnegie Mellon University.

DM-0004410