The Opportunities and Complexities of Applying Commercial-Off-the-Shelf Components

Lisa Brownsword, David Carney, and Tricia Oberndorf
Software Engineering Institute

Government acquisition policies for software-intensive systems emphasize the use of commercial-off-the-shelf (COTS) products. On the surface, "the COTS solution" appears straightforward. In actuality, many projects find its use less than straightforward. This article provides acquisition managers and policy makers with a basic understanding of how developing systems with COTS products is different and why and what new COTS capabilities are being identified.

Government acquisition policies for software-intensive systems now emphasize the use of existing commercial products. Requests for Proposals often require the use of specific COTS products and sometimes specify the amounts to be used. As systems are reengineered, many include the use of COTS products. And as government budgets shrink and the desire for increasingly complex systems continues, there is rising interest in leveraging the use of commercially available products when possible.

Although on the surface "the COTS solution" appears straightforward and compelling, projects that apply COTS find its use less than straightforward. Rather, they encounter significant new trade-offs and issues. Applying COTS products is not merely a technical matter for system integrators. It has a profound impact on business, acquisition, and management practices, and organizational structures. Compounding the problem is the limited experience and guidance currently available on how to effectively approach system development with commercial components.

The Software Engineering Institute (SEI), along with other key organizations associated with the government and civil agencies, is creating and assembling best practice guidance for acquisition and program managers, integrators, and testers through case studies, hands-on support, and analysis. This article is one of several venues the SEI is leveraging to provide acquisition managers and policy makers with an understanding of what is different in the development of

systems with COTS products and why and what new capabilities are being identified. Due to the brevity of this article, the discussion is limited to a few essential aspects of the differences and capabilities of COTS products.

What Is Different with COTS-Based Systems?

COTS products can be applied to a spectrum of systems. At one end of the spectrum are nearly packaged software solutions, such as Microsoft Office or Common Desktop Environment, that require no integration with other components. These systems map well to the needs and operations of the government.

Further along the spectrum are COTS products that support the information management domain, such as Oracle or Sybase. These systems typically consist of both COTS products and custom components, with COTS products making up the majority of the system. Depending on how well the COTS products and custom components fit together, a small to moderate amount of customization is usually required to enable them to work cooperatively.

At the other end of the spectrum, there are systems composed of a complex mix of commercial and noncommercial products that provide large-scale functionality that is otherwise not available. Such systems typically require large amounts of "glue" code to integrate the set of components. These systems are typically in the embedded, real-time, or safety-critical domains.

Using COTS products for applications at the packaged software solutions

end of the spectrum is relatively straightforward; however, using them for complex systems further along the spectrum is not. This article focuses on the issues and complications that arise when constructing complex systems with COTS products.

Fundamental Paradigm Change

Traditionally, organizations develop systems from scratch with control over all or most of the pieces. They

- collect and define requirements.
- identify an architecture to satisfy the requirements.
- design in detail individual subsystems to fit within the architecture.
- code, test, and debug modules to meet the specified requirements.
- integrate sets of modules and subsystems into the complete system.

With the use of COTS as components for a system, a fundamental change occurs: an organization now *composes* the system from building blocks that may or may not (generally do not) work cooperatively directly out of the box. The organization will require skilled engineering expertise to determine how to make a set of components work cooperatively—and at what cost.

This fundamental shift from development to composition causes numerous technical, organizational, management, and business changes. Some of these changes are obvious, whereas others are quite subtle, but if not addressed, either type of change can cause severe problems for the project. Consequently, organizations may have to modify their

procedures and structures and, in some cases, create entirely new procedures.

Impact on Typical Lifecycle Activities

Regardless of which lifecycle model an organization uses (waterfall, spiral, or iterative), they perform requirements, architecture, detailed design, code, test, and system integration activities. The use of COTS products has a pervasive impact on all lifecycle activities. This is illustrated by briefly examining the impact to requirements, testing, and maintenance activities.

Requirements describe the desired system behavior and capability with a set of specified conditions. For a COTS-based system, the specified requirements must be sufficiently flexible to accommodate a variety of available commercial products and their associated fluctuations over time. To write such requirements, the author must know enough about the commercial marketplace to describe functional features for which commercial products exist.

There is a critical relationship among technology and product selection, requirement specification, and architecture definition. If you define your architecture to fulfill your requirements and then select your COTS products, you may have only a few or no available products that fit within the chosen architecture. Pragmatically, three essential elements (requirements, architecture, and product selection) must be worked in parallel with constant trade-offs among them.

As the **testing** of COTS-based systems is considered, you must determine what levels of testing are possible or needed. A COTS product is a "black box" and therefore changes the nature of testing. A system may use only a partial set of features of a given COTS product. Should you test only the features used in the system? How do you test for failures in used features that may have abnormal behavior due to unknown dependencies between the used and unused features of a COTS product?

Maintenance also changes in fundamental ways—it is no longer solely concerned with fixing existing behavior or

incorporating new mission needs. Vendors update their COTS products on their own schedules and at differing intervals. Also, a vendor may elect to eliminate, change, add, or combine features for a release. Updates to one COTS product, such as new file formats or naming convention changes, can have unforeseen consequences for other COTS products in the system. To further complicate maintenance, all COTS products require continual attention to license expirations and changes. All these events routinely occur. All these activities may (and typically do) start well *before* an organization fields the system or major upgrade. Pragmatically, the distinction between development and maintenance all but disappears.

Emergence of Nontypical Activities

We can view the commercial marketplace as a continuous "product conveyor belt"—the marketplace constantly adds new products and technology to the belt, existing products evolve through continuous upgrades, and vendors remove products from the marketplace. The government has limited influence (and no direct control) over the speed, content, or variety of products on the product belt.

Consumers, such as the government, must constantly keep abreast of the state of the product belt. This requires new activities (with associated resource requirements) in the area of technology and product evaluation. Consumers must identify potential technologies and products, qualify candidates for fit within their system, and perform trade-off analysis between competing technologies and products. An organization must *continuously* perform the entire process of monitoring, evaluating, qualifying, and analyzing the impact of technology and products given the constant changes within the commercial marketplace. We should add that technology and product awareness and evaluation are not activities that the government can merely relegate to its contractors. The government must also have such a capability if it is to specify and manage its systems wisely.

Assembling COTS products also presents new difficulties. Although software COTS products are attempting to simulate the "plug-and-play" capability of the hardware world, in today's reality, software COTS products seldom plug into anything easily. Most products require some amount of adaptation to work harmoniously with the other commercial or custom components in the system. The typical solution is to adapt each software COTS product through the use of "wrappers," "bridges," or other "glueware." It is important to note that adaptation does *not* imply modification of the COTS product. However, adaptation can be a complex activity that requires technical expertise at the detailed system and specific COTS component levels. Adaptation must take into account the interactions among custom components, COTS products, any non-developmental item components, any legacy code, and the architecture, including infrastructure and middleware elements. This adaptation process has a cost—a potentially high one.

What Should an Acquisition or Program Manager Do to Get Started Using COTS?

Applying a COTS solution requires the government to create and maintain new competencies. We have alluded to a number of essential capabilities throughout the article. The following sections identify a number of actions to establish an effective infrastructure for the use of COTS products. These actions are not intended as a road map or to be all-inclusive. Rather, they are a set of practical actions to help organizations *start* to develop the necessary knowledge and experience base. We recommend that organizations begin now—ideally before the first (or next) COTS-based system development, reengineering, or maintenance project.

Know the Regulations
There are various policies, regulations, and directives relative to the general use of commercial products. Policies also exist concerning the use of specific COTS products such as the Distributed Information Infrastructure Com-

mon Operating Environment. Understand what policies and directives apply and how they apply to your particular systems. Situations or directives may change. Therefore, an organization should have a "regulations guru" available whose ongoing work is to remain informed of the various government regulations and their impact to the organization's systems.

Know Your Marketplace
The COTS marketplace is huge and continually changes. Determine which subsets of the marketplace are relevant to your systems. Develop dedicated resources to become conversant with the available and emerging COTS technologies and products, and determine their impact for your applications.

Know How to Evaluate Technologies and Products
Determining which products and technologies are most appropriate for a given system requires more than a market survey based on marketing literature and vendor demonstrations. Know how to develop evaluation criteria, conduct a satisfactory evaluation, and select viable technologies and products based on your criteria. Determine the amount of time to allocate for evaluation during the acquisition. Leverage previous projects to experiment with developing the expertise required.

Know How to Develop Requirements
It is vital to understand how to specify requirements; this strikes the optimum balance between desired user functionality and the available COTS product. Know how to make trade-offs between COTS products, your architecture, and your requirements. Again, leverage previous projects to experiment with developing the required expertise.

Know How to Manage System and COTS Product Evolution
Learn how the development and maintenance of your systems will need to change as a result of the continual release of COTS product updates. Some sample areas to investigate include scheduling of COTS product updates

into your baseline, impact analysis to determine potential interactions and changes to existing components, impact analysis to the operations of the fielded system, and identification and management of licensing issues for your intended COTS products.

Determine How to Build Your Business Case
Although the motivation for the use of COTS products for many organizations is cost savings, an organization should address the many business unknowns prior to making that determination. How are the costs of both the initial and recurring adaptation throughout maintenance determined? How should a program manager make the business case if the total lifecycle cost is higher?

Develop a Metrics Database to Determine Your Business Case
Currently, there is little data on the cost, schedule, or quality benefits of COTS-based systems. Begin collecting the data needed to develop a realistic business case. Such data might include cost, time distribution across lifecycle activities, defects after the system is fielded, or efficiencies gained or lost in field operations.

Final Remarks
Even if an organization obtains some parts from commercial sources, a COTS-based system is still a system with requirements. Only the people who pay for, maintain, and use a COTS-based system are concerned about the quality of the system—vendors are not. Organizations must still design, assemble, test, manage, and maintain the system. There is no magic. There is no COTS "silver bullet." The government's responsibility for its systems is not eliminated or reduced by a reliance on COTS products.

COTS systems require acquisition and program managers, policy makers, systems integrators, and system designers to become smart consumers by understanding the business, management, organizational, and technical implications of applying COTS products to system development or reengineering. The worst thing you can do is treat the

shift toward the use of COTS products as merely a change in technology. ◆

About the Authors
Lisa Brownsword is a member of the technical staff in the Dynamic Systems program at the SEI. Before the SEI, she worked for Computer Sciences Corporation in support of the NASA/Goddard Software Engineering Laboratory. Prior to that, at Rational Software Corporation, she provided consulting to managers and technical practitioners in the use of software engineering practices, including architecture-centered development, product lines, object technology, Ada, and computer-aided software engineering (CASE).

> Software Engineering Institute
> Carnegie Mellon University
> 4301 Wilson Boulevard, Suite 902
> Arlington, VA 22203
> Voice: 703-908-8203
> Fax: 703-908-9317
> E-mail: llb@sei.cmu.edu

David Carney is a member of the technical staff in the Dynamic Systems program at the SEI. Before joining the SEI, he was on the staff of the Institute for Defense Analysis in Alexandria, Va., where he worked with the Software Technology for Adaptable, Reliable Systems program and with the NATO Special Working Group on the Ada Programming Support Environment. Prior to that, he was employed at Intermetrics, Inc., where he worked on the Ada Integrated Environment project.

> Software Engineering Institute
> Carnegie Mellon University
> Pittsburgh, PA 15213
> Voice: 412-268-6525
> Fax: 412-268-5758
> E-mail: djc@sei.cmu.edu

Tricia Oberndorf is a member of the technical staff at the SEI. She is a part of the Dynamic Systems program and concentrates on the investigation of integration and open systems issues. She has investigated a number of other integration and open systems questions in the context of CASE environments and other kinds of systems. Prior to the SEI, she worked for the Navy for more than 19 years.

> Software Engineering Institute
> Carnegie Mellon University
> Pittsburgh, PA 15213
> Voice: 412-268-6138
> Fax: 412-268-5758
> E-mail: po@sei.cmu.edu