

COTS Evaluation in the Real World

David Carney



In my last column, I covered the overall topic of evaluating commercial off-the-shelf (COTS) products. To summarize the major points:

- “COTS evaluation” can have various meanings. For our purposes, the intended meaning is that COTS evaluation is a process to decide whether to select one product for use in a given context.
- The evaluation process is pervasive. That is, evaluation is not restricted simply to the moment when an assessment of a product is made, but is operative at many points, e.g., when we define our selection criteria, when we perform vendor surveys or market research, and so forth.

I also suggested that there are three large-scale tasks that occur when any COTS evaluation is performed:

1. Plan the evaluation.
2. Design the evaluation instrument.
3. Apply the evaluation instrument.

In this column, I will examine two actual organizations that carry out COTS evaluation processes, to see how this abstract notion of COTS evaluation fits with real-world experiences.

The central idea: the importance of context

Before I do so, however, I must return to a thought that was touched on in the last column, and whose importance I will now stress. Central to COTS evaluation—in the particular sense of it as a process rooted in decision making—is the importance of *context*. By context I mean all of the factors and constraints (functional, technical, platform, business issues, and so forth) that exist before a COTS product is chosen. “Context” includes everything with which and against which the COTS product must harmonize, conform, and operate; it is the basis on which we develop evaluation criteria to assess the product.

This notion seems rather simple, and perhaps obvious, but it has some rather far-reaching implications. For one thing, in using COTS products in complex systems, it is difficult to continue with our traditional requirements-driven processes. COTS products are typically written to the vendor's own predictions and expectations of what will have market success. The requirements of our particular system, i.e., the one that will incorporate those products, are not known to the vendor (nor would they necessarily be of interest to the vendor even if known).

So instead of a set of hard "must haves" by which we will judge a product, we have a much more fluid collection of features, some mandatory, some strongly desirable, some merely "nice to have"; in short, this collection of features provides the *context*, the source of the criteria by which we will decide whether a given COTS product is sufficient for our needs.

Another implication is that this context can be very wide, containing both technical and business-oriented issues, and the kind of tradeoffs we will make will necessarily mix apples and grapefruit. We must therefore find ways in the evaluation process to reconcile constraints that compete along different axes of interest. For instance, we will often need to make the choice between product A, which has wonderful throughput but whose vendor seems to be moving out of this particular market niche; and product B, whose vendor we know and trust, but whose product is slower (not disastrously slow, perhaps, but certainly creaky by comparison). So should we buy from trusted vendors, but with near-obsolete products? Or should we choose a bleeding edge of technology, knowing that it comes from three guys in a basement?

Finally, since this evaluation context will in some ways replace a complete set of requirements, it is very possible that novel or unexpected features of COTS products will actually change our idea of the system as it is being built. There are numerous instances, one of which I will describe presently, in which the features of emerging products can bring about changes in the design, or even the overall architecture, of a COTS-intensive system. This is indeed a radical situation for people who consider that an architecture (or design, for those who think they are the same) must be established at the beginning and that COTS products must be found to conform to it.

These implications are quite real. If we make a significant policy decision to use pre-existing pieces (i.e., if we mandate that our systems make extensive use of COTS products), then we must relinquish our grip on the details, be willing to suffer the (sometimes chaotic) fluctuations of the marketplace, and be content to let some parts of our systems be fashioned by the commercial forces that drive that marketplace.

A glimpse at current practice

We now will consider how two very different organizations deal with the problem of evaluating COTS products. One organization ("A") is a major contractor that has

provided numerous systems for the government. The other organization (“B”) is a private corporation in the domain of business services and financial transactions. While the two use very different approaches to COTS evaluation, we still observe our abstract process ideas underlying both. And we also learn that these two different approaches corroborate our notion that whatever the actual process used, it is driven by context.

Organization A

This organization is building a large information system for a government agency. Lest *large* be misunderstood, the system will incorporate over 60 commercial components, many of them based on such evolving technologies as Web browsers, Java-based products, and other middleware products. The system, now partially fielded, serves several thousand users and is deployed worldwide on a variety of platforms.

The domain of this system, that of Web-based information systems using browsers, Java, and its derivative technologies, is scarcely half a decade old and is still growing and evolving with lightning speed. And as fast as the marketplace of products is changing, the marketplace of ideas is changing even faster. So competition among companies (some barely deserving of the name) is cutthroat, with product releases occurring rapidly. In addition, there is little stability even regarding which products should perform which functions (i.e., these products often lack a condition called “product integrity”).

For this organization, and in the context of this system, there is a fundamental need for flexibility. Since it is known that the system makes use of evolving technologies, it is understood that new elements will be refreshed and replaced often, and that design decisions (and thus COTS product choices) are subject to frequent reconsideration. Therefore, COTS evaluation for this organization is flavored accordingly.

The context for evaluation is primarily technical: the aggregate collection of factors (e.g., interfaces, standards) that permit or prevent interoperation among components. Thus, while individual products are assessed for their individual functionalities, they are assessed as much to see if they interoperate and cooperate with the other COTS components in the system.

The following indicates how our abstract process steps are instantiated for COTS evaluation by this organization:

Plan the evaluation. The planning for evaluation is generally not rigorous or exhaustive, since the expected lifetime of any specific product within the system is relatively short. Planning is therefore opportunistic. Given the number of COTS products used and the number of potential candidates, a careful, methodical approach is simply not feasible.

Design the evaluation instrument. What we have abstractly termed the “evaluation instrument” is weighted with criteria about compatibility. In fact, the “evaluation instrument” is really the existing system with which the candidate product must operate.

Apply the evaluation instrument. Assessing products is generally done through prototyping and through installation of the product into the existing system context.

The evaluation process itself is as flexible as possible, and serendipity is not considered a bad thing. The unexpected appearance of new capabilities (whether within a product or through more general technology trends) can trigger reassessment of decisions made earlier.

One thing we observe in this style of system development (other than that it is frightening to many people!) is the mutual influences that evaluation and system design have on each other in a COTS-intensive system. I will consider this topic in greater detail in my next column.

Organization B

This organization is a large financial institution that purchases a large number of COTS products each year. The business processes of this industry are relatively stable, and while heavily dependent on data processing, are usually not influenced by any rapidly changing technological trends.

COTS evaluation, as performed by this organization, is most often done to choose products to modernize existing capabilities. While there is interaction among all of the organization's systems, new COTS products are not perceived as components within a single large system. Hence, while compatibility with the existing platforms and infrastructure is significant, new products might be considered even if they depart from the existing technical infrastructure.

The following shows how our abstract process steps are instantiated:

- *Plan the evaluation.* The organization has a large set of procedures about how to prepare evaluation plans, how to conduct product searches, how to conduct vendor assessments, and so forth. Strict guidelines exist that require particular planning activities depending on business importance of the product, expected cost, and similar factors.
- *Design the evaluation instrument.* Choices about evaluation criteria are strongly weighted toward business factors: vendor stability, reports about vendor's market share, and satisfaction reports from other users of the product
- *Apply the evaluation instrument.* There is little prototyping. Often the actual assessment of product capabilities is limited to the vendor's presentation. The hands-on technical evaluation done in house is largely focused on platform compatibility with existing systems in the organization.

Last thoughts

The ways that these two organizations perform COTS evaluation are radically different, which should not be surprising given how different their contexts are. Organization A's evaluation plans are necessarily opportunistic, while B's are rooted in method.

Organization B's "evaluation instrument," especially in its choices about evaluation criteria, is weighted toward business factors, while A's are weighted toward technical compatibility. And when actually assessing products, Organization A does extensive prototyping while B does comparatively little in-house product examination.

These differences reflect very different circumstances, and in no way imply rightness or wrongness of either instantiated process. But both are examples of how a commercial bias will affect the evaluation process, and ultimately the entire process of system construction.

In the next column, I will focus on the kind of evaluation process found in Organization A, since it is here that we can observe the point noted above, namely, that there is a very strong mutual influence between designing a COTS-based system and evaluating candidate products for that system. Stay tuned.

About the author

David Carney is a member of the technical staff in the Dynamic Systems Program at the SEI. Before coming to 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 Ada Programming Support Environment. Before that, he was employed at Intermetrics, Inc., where he worked on the Ada Integrated Environment project.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

SM IDEAL, Interim Profile, Personal Software Process, PSP, SCE, Team Software Process, and TSP are service marks of Carnegie Mellon University.

® Capability Maturity Model, Capability Maturity Modeling, CERT, and CMM are registered in the U.S. Patent and Trademark Office.