Evaluating Components in the Context of Use

Steven R. Haynes  
Penn State University  
shaynes@ist.psu.edu

Sandeep Purao  
Penn State University  
spurao@ist.psu.edu

Julian R. Winter  
Penn State University  
jwinter@cse.psu.edu

Abstract

A number of methods have been proposed for selecting and evaluating commercial-off-the-shelf (COTS) component-based systems. Few methods, however, focus on post-implementation evaluation of these systems in the context of use. This paper reports on the use of one such method that employs scenarios as the central unit of analysis. The method is inspired by immersive, ethnographic techniques and informed by work in scenario-based design. In this paper, we describe the development of the method and its application for evaluating a complex, COTS-based system for product lifecycle management (PLM).

1. Introduction

In spite of considerable work developing evaluation processes and selection criteria, realizing effective systems using commercial-off-the-shelf (COTS) components continues to be a challenge [1,13]. In particular, post-implementation evaluation is a challenge that requires useful techniques because the impacts of systems in use are difficult to assess [4] and can evolve significantly from original requirements [6]. Post-implementation evaluation may be summative, assessing whether stated development goals have been met, or formative, suggesting strategies for future design changes [2]. Evaluations are especially difficult for systems developed with COTS components because organizations tend to focus on concerns related to component attributes, component features, and component integration, sometimes ignoring basic user concerns [10].

In this paper, we report on a method aimed at evaluating a COTS-based product lifecycle management (PLM) system deployed at a U.S. Marine Corps Program Management (PM) Office. The system is designed to support the collaborative work of civilians and Marines in their use, maintenance and evolution of a Light Armored Vehicle (LAV). Over the last year we conducted an assessment of the system, which is used by virtually all PM office personnel.

This paper describes the evaluation method including the rationale that lead to its development, reports results on its efficacy, and shares lessons learned from its application. Our experience in applying the method so far suggests that it provides an appropriate mix of grounded yet tractable evaluation with scenarios as an effective unit of analysis. The results further suggest that the method aids in understanding COTS capabilities and dysfunctions in the context of concrete work situations.

2. COTS Evaluation Practices

Prior research into COTS assessment exhibits a clear focus on pre-implementation evaluation of components, arguably justifiable given the complexity of many selection decisions. Using a questionnaire-based evaluation of six competing products, Maiden et al. [10] present lessons learned and recommendations for selecting complex COTS software components. First, they suggest that detailed requirements be identified early to help reduce the set of components for consideration, which can be subsequently evaluated in parallel with detailed requirements acquisition and evaluation. Second, they suggest constructing prototypes and testing these against identified scenarios to explore how different alternatives support scenario completion. Third, they suggest that requirements be measurable to enable product selection.

Another study [9] develops what the authors describe as a systematic, repeatable and requirements-driven COTS selection process with explicit definitions of tasks, and an incremental, hierarchical and detailed definition of evaluation criteria. Yet another [11] provides an example that begins with high-level requirements and proceeds at increasing levels of detail to identify, integrate, and implement the COTS components that make up a new system. Based on problems discovered in early versions of the process, they suggest an updated evaluation method focused on merging requirements analysis and COTS selection as a single activity strand and explicitly recommend adapting the COTS selection process to consider both technical and non-technical criteria.

More recently, Torchiano et al., [12] suggest a set of evaluative objectives and qualitative attributes to assess the appropriateness of COTS components relative to a given set of requirements. Their suggestions allow for the possibility that under-defined requirements may themselves undergo change driven by the capabilities of
As the focal point in a reflective design [5] process, scenarios provide a basis for evaluating decisions within the design space in a manner that is grounded in the actual use of the system, rather than in proposed or idealized system use. The narrative format of scenarios also provides a better cognitive fit between the analyst and the assessment task, following Carroll’s [2] claim that stories are a more natural, flexible, and easier to use lens than more structured, formal techniques.

We develop our method by extending scenario-based design for scenario-based evaluation. Figure 1 outlines the conceptual model underlying our evaluation method.

![Evaluation model](image)

**Figure 1. Evaluation model**

Scenarios can be used to move from an evaluative to a formative perspective, that is, for the purpose of understanding how improvements can be made to overcome identified obstacles. Consider, for example, the claim “The task workflow component must be made more flexible” suggesting customization of the underlying component, in this case, the Microsoft Exchange Server workflow subsystem. The method has the potential to translate, through claims analysis, the narrative of the scenario into concrete assessments of the system’s contributions, and specifically, how each component facilitates or hinders completion of scenarios of use.

### 4. Setting and Procedures

The scenario-based evaluation method outlined above was applied for assessment of an integrated digital environment (IDE) implemented to facilitate collaborative engineering across functions at the offices of a U.S. Marine Corps program manager (PM). Some of the important features of the IDE are its communications (e-mail), workflow, document management, project management, collaborative engineering, and performance reporting functionality. The PM in this case manages a fleet of about 800 vehicles deployed worldwide in a variety of configurations and tasked with a range of missions. Much of the work of the PM consists of monitoring vehicle health and field performance, analyzing opportunities to enhance performance,

COTS components being considered. Alves & Finkelstein [7] make a similar argument acknowledging the fact that requirements need to be flexible to ensure that plausible COTS components are not excluded from consideration. They use techniques from goal-oriented requirements engineering to allow evolution of requirements during the system life cycle. Similar to [10], which uses test cases, they suggest that goals be defined with the help of use cases that are used as benchmarks throughout the component selection and system development process.

These prior efforts show a clear focus on pre-implementation evaluation of COTS components, a focus that appears justifiable given the complexity of the selection decision. Recent field studies, however, show that the lifecycle costs of COTS-based systems may actually exceed that of custom developed software [13], which points to an underlying need for ongoing tracking of COTS-based systems in production to assess their contributions to the organization.

Research in COTS evaluation reveals several recurring themes. First, system requirements are seen as an essential input to component selection process though it is acknowledged that requirements can evolve based on the COTS components selected. Second, test cases, use cases, and scenarios appear repeatedly as the unit of analysis researchers suggest as the basis for component selection. Gaps in prior research reveal the need to complement evaluation and selection with post-implementation evaluation techniques for COTS-based systems in production. Current methods provide solutions that are either too abstract or too focused on the technology, as opposed to focusing on the concrete uses of components in their intended work context. Existing methods also do not provide adequate linkage between the measurable outcomes of evaluation and the organizational priorities that technology is intended to support.

### 3. Evaluating COTS in the Context of Use

To address these issues, we developed an evaluation method leveraging concepts from scenario-based design and claims analysis [2,3], which grounds evaluation in concrete use scenarios. Scenarios take the form of loosely structured narratives that describe details of a user interaction with a system. A scenario includes actors, settings, goals, and, importantly, claims about the efficacy of proposed designs [2]. Actors may be individuals but usually are represented as prototypical roles. Setting describes the context in which the scenario occurs. Goals give the motivation behind performing a scenario. Claims explore how a system or feature will support the scenario and are the mechanism by which designers reflect on the trade-offs of options with respect to this support [3].
managing maintenance, and design and implementation of engineering upgrades. The role of the IDE is to support the PM function by providing an integrated environment for information management related to these tasks.

The evaluation method was operationalized in the form of interviews that were conducted at the PM’s office. A total of 26 interviews were conducted with a view to understanding how and to what degree the existing system and components embedded therein support organizational goals and tasks. The interview participants represented personnel at a range of seniorities with a range of functions including engineering, logistics, and business operations. Interviews were conducted following a semi-structured guide provided in the figure below.

**General**

*What is your position?*

*Describe your different roles in the Program Management function for LAV.*

**Current systems use and processes**

*Describe your use of the IDE in terms of scenarios.*

*How do these scenarios contribute to the mission of the PM LAV?*

*What information and functions of the IDE do you find most useful?*

**Prospective systems use and processes**

*What information needs are not currently being met by the IDE?*

*How could any of these unmet information needs be met by the IDE?*

**Figure 2. Study Interview Guide**

The interviews were taped and transcribed. The qualitative analysis package **Atlas.ti** (www.atlasti.com) was used to assist with transcript coding and review. We looked for participant descriptions of their use scenarios and the follow-on discussions of the contributions derived from system support for these scenarios.

### 5. Results

The analysis so far has yielded 43 scenarios across nine different functional modules of the IDE. Of these 43 scenarios, 27 were reported as current scenarios of use and 16 as envisioned scenarios. Many scenarios were discussed by multiple interview participants. Several of the scenarios may be classified as routine office procedures, others facilitated collaboration across different roles following predetermined organizational processes, and others facilitated information sharing with use of repositories in an *ad hoc* manner. The total number of claims was 242 of which 134 were positive claims and 108 negative. The average number of claims associated with each scenario was 5.6 with a range of from 0 to 22.

Many obstacles to effective scenario support were reported among these claims.

To further understand how scenarios are supported or hindered by components embedded in the system, claims were linked to underlying IDE features. Figure 3 shows a fragment of a scenario-claims-component network that shows the mapping between claims, the underlying system capabilities, and the components that realize these capabilities. The technique helps illuminate components responsible for the most negative claims as well as helping to understand attributes of the IDE components most responsible for positive perceptions among system users. Linking scenarios and claims to COTS components is one of the more labor-intensive tasks in the method. It involves tracing components involved in the execution of the scenario, and determining the linkage between an individual claim and the component(s) supporting that part of the scenario responsible for the claim. This mapping and an aggregated summary of the claims related to the component provide a qualitative, multi-dimensional (technical, psychological, social) evaluation of the component’s performance and suggests potential re-design, user training, or changes to organizational processes to better fit the component’s capabilities.

**Figure 3. Scenario-Claim-Component network**

For example, in the network fragment in Figure 3 above, the claim regarding inflexible support for the scenario (Can’t modify orders) suggests an area where customization of a specific COTS component might improve support for the task. The example illustrates a one-to-many relationship between scenarios and claims, and a many-to-one relationship between claims and component. Further results cannot be included here due to space constraints but will be discussed at the workshop.

### 6. Discussion

To-date, most COTS component evaluation methods have focused on technique for selection of components as part of planned systems. The objective of the work reported here is development of a method to help evaluate COTS-based systems that are already implemented and deployed in their intended environment. The purpose of
this post-implementation evaluation method is to understand how components embedded in the system contribute to or detract from concrete scenarios of use that conform to organizational objectives, and to drive design enhancements that can be traced to specific components embedded in the implemented system. The scenario-claims-component network we demonstrate is an integral part of our method, which directly contributes to the objective of post-implementation evaluation of specific COTS components in the context of their use.

During the application of the method, we have found that articulation of scenarios and claims can act as focal points that are evocative without being provocative. Since scenarios and claims are the result of interaction with system users, not the product of evaluator or other IT member analyses, we have found that focus group interactions are channeled into resolving conflicting claims, clarifying the meaning of ambiguous claims (e.g., system support for the scenario decreases task efficiency), and identifying new claims not collected in one-on-one interview sessions, rather than on correcting potential IT team misinterpretation or missed analyses.

A number of challenges emerged from the study. Among the most significant is transforming scenario and claims data into recommendations for system developers and integrators responsible for managing COTS component selection, implementation, and maintenance tasks. Unpacking how specific system features and feature deficiencies contribute to unrealized system contributions requires analysts with significant knowledge of the system’s underlying architecture. Even with this knowledge, relating qualitative remarks that are sometimes very general and often vague to specifically what can be done to address the deficiency requires inferential leaps that can be difficult to justify. The fragment shown in Figure 3 that connects claims to specific components represents an instance of how this knowledge can be articulated.

The relative expense of collecting and analyzing the data from qualitative interviews continues to be a potential barrier to method adoption. Based on our experience so far, the benefits of this rich data outweigh the costs. As the expense of maintaining COTS-based systems begins to exceed that of custom systems [13] these costs may become easier to justify. On-going, post-implementation evaluations that help track where an underlying component is supporting scenarios with the greatest contribution, and where a component fails to support high-value scenarios, are useful metrics for measuring ongoing system performance and return on investment.

7. Concluding Remarks

We have described an approach to COTS-based system evaluation that represents a middle-ground between high-level taxonomies of evaluation topics and low-level gap-fit assessments of COTS features. The scenario-based evaluation method described here targets concrete instances of use as the primary level of analysis and considers claims associated with system support for the instance of use. The method shows promise in that it is relatively easy to learn and apply, is able to account for the range of phenomena that impact a COTS component’s ability to support a domain requirement, and is capable of being translated into prescriptions for action.

8. References