NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

A Process for COTS Software Product Evaluation

Comella-Dorda, S.; Dean, John; Morris, E.; Oberndorf, P.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Proceedings of the 1st International Conference on COTS-Based Software System, 2002

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=ec6c6ca6-31df-4c00-8d50-79cfe650c5b8
https://publications-cnrc.canada.ca/fra/voir/objet/?id=ec6c6ca6-31df-4c00-8d50-79cfe650c5b8

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

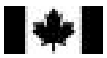
L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Canada



National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC-CNRC

*A Process for COTS Software Product Evaluation**

Santiago Comella-Dorda, John Dean, Edwin Morris, and Patricia Oberndorf

February 2002

* A process for COTS software product evaluation. Proceedings of the 1st International Conference on COTS-Based Software System, Orlando, FL. February 4-6, 2002. pp. 86-96. NRC 44925. (LNCS 2255, Springer).

Copyright 2002 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

A Process for COTS Software Product Evaluation

Santiago Comella-Dorda[2], John Dean[1], Edwin Morris[2] , and Patricia Oberndorf[2]

[1] Building M-50 Montreal Road, National Research Council Canada, Ottawa, Ontario, Canada, K1A 0R6
John.Dean@nrc.ca
[2] CBS Initiative, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA USA,
(po, ejm, scd)@sei.cmu.edu

Abstract. The growing use of commercial products in large systems makes evaluation and selection of appropriate products an increasingly essential activity. However, many organizations struggle in their attempts to select an appropriate product for use in systems. As part of a cooperative effort, the Software Engineering Institute (SEI) and the National Research Council Canada (NRC) have defined a tailorable software product evaluation process that can support organizations in making carefully reasoned and sound product decisions. This paper describes that process.

1 Introduction

Many organizations find themselves faced with the prospect of constructing major software systems from commercial off-the-shelf (COTS) products[1]. An essential part of such an endeavor is evaluating the commercial products that are available to determine their suitability for use in the particular system. Yet, as we look at the experiences of these organizations, we find one of the hard lessons of COTS product evaluation: reasonable people doing reasonable things still have problems that are traceable to the quality of their evaluation process. Among the common evaluation mistakes we have seen are:

- Inadequate level of effort
- Neglecting to re-evaluate new versions or releases
- Use of "best of breed" lists that do not reflect the characteristics of the system
- Limited stakeholder involvement
- No hands-on experimentation

[1] A COTS product is a product that is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor, who retains the intellectual property rights; available in multiple, identical copies; and used without modification of the internals. This definition is by design imprecise, since the COTS market continues to redefine itself. We do not intend this definition as an "acid test", but rather to provide the general essence of a COTS product.

In response to these and other problems, the Software Engineering Institute (SEI) and the National Research Council Canada (NRC) have co-developed a COTS software product evaluation process that is tailorable to suit the needs of a variety of projects. This evaluation process addresses the examination of COTS products for the purpose of determining their fitness for use in a system.

2 An Evaluation Process

We have seen many cases where projects are told to pick a particular product because the vendor offered a good deal (or it was on a list, or the boss wanted it). We believe that consistently good evaluation results can only be achieved by following a high quality and consistent evaluation process. This does not mean that each evaluation activity requires a highly complex, exquisitely documented process (although sometimes they do), but if you do not follow some kind of consistent process, it is likely that the quality of your results will vary.

The high level process we describe is flexible and amenable to many specific process implementations. It consists of four basic elements:

- **P**lanning the evaluation
- **E**stablishing the criteria
- **C**ollecting the data
- **A**nalyzing the data.

The process, called PECA, begins with initial planning for an evaluation of a COTS product (or products) and concludes with a recommendation to the decision-maker. The decision itself is not considered as part of the evaluation process -- the aim of the process is to provide all of the information necessary for a decision to be made. PECA is in part derived from ISO 14598 [1]. Where our experience differed from ISO 14598, we freely changed the process to fit our needs.

As illustrated in Figure 1, the elements in the PECA process are not always executed sequentially. Evaluation events, such as a need for new criteria to distinguish products, unexpected discoveries that lead to the start of a new iteration, or inadequacy of collected data, will direct process flow through one of the process elements as needed. One of the hallmarks of the PECA process is this flexibility to accommodate the realities of COTS-based systems.

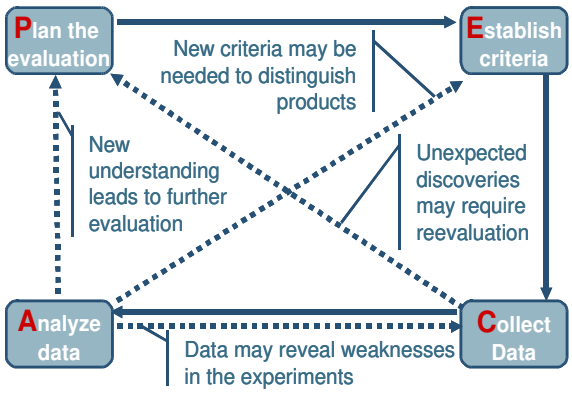


Fig 1: PECA – A Recommended

The PECA process is intended to be tailored by each organization to fit its particular needs. PECA is

flexible enough to be used within many organizations and with many COTS-based development processes.

Finally, a successful COTS evaluation relies on more than just a process. In addition you will need to employ a set of techniques that allow you to plan, establish criteria, and collect and analyze data effectively. For example, the popular GQM technique [2] can be used to establish COTS evaluation criteria. This paper will not devote significant space to techniques. More complete coverage of techniques is available in a COTS product evaluation tutorial developed by the authors.

3 The PECA Process Examined

The main section of this paper to provides detail about the four basic elements of the PECA process. As you consider these elements, keep in mind that PECA assumes a highly contextual evaluation of a COTS product. This implies that part of a PECA evaluation will be conducted in concert with evaluations of other COTS products that are also being considered for use in the system. A PECA evaluation is therefore a complex activity in which individual products are not evaluated in isolation.

Note also that PECA considers the fitness of a product for use to involve more than just meeting technical criteria. Criteria can also include such concerns as the fitness of the vendor (e.g., reputation, financial health), the technological direction of the marketplace, and the expectations placed on support staff.

This breadth relies on a wide range of inputs. Two obvious inputs are the set of products that will be considered and the system requirements (both functional and nonfunctional) that must be met. However, system requirements alone are normally not sufficient for making an appropriate choice from among the set of products. They often fail to address many important characteristics of COTS products and vendors, such as underlying technology, quality of reputation, and support services offered.

The expectations held by stakeholders are another important input. Such expectations are imperfectly captured as system requirements, yet often determine the eventual success of the COTS product in the system. In addition, the use of COTS products may introduce an entirely new set of stakeholders. Another set of inputs are system decisions that have already been made regarding system architecture and design, other system components, and development and maintenance processes to be supported. These will constrain the COTS product selection.

3.1 Planning the Evaluation

Planning for each COTS evaluation is different, since the evaluation may involve both different types of products (from simple to extremely complex) and different system expectations placed on the product (from trivial to highly demanding).

Forming the evaluation team. The importance of an effective team for a successful evaluation should not be underestimated. Unfortunately, there are situations where the most junior engineer, with little support from others, is assigned to evaluate products. In most cases, a lone engineer – even a senior engineer – does not have the range of skills necessary to perform a broad-based COTS evaluation. Most evaluation

teams should include technical experts, domain experts, contracts personnel, business analysts, security professionals, maintenance staff, and various end users. And, as with any team, a good balance of power is important, so no single individual can bias the results toward his personal preferences.

Creating a charter. The evaluation team creates a charter that defines the scope and constraints of the evaluation. The charter includes a statement of the evaluation goals, the names and roles of the team members, a commitment statement of from both the evaluators and their management, a summary of factors that limit selection, and a summary of decisions already been made.

Identifying stakeholders. The stakeholders for the entire system may have already been identified, and some may be included on the evaluation team, but each COTS evaluation entails its own, often unique, set of stakeholders. Evaluation stakeholders are those individuals or groups with vested interest in the results of a COTS evaluation or on whom the selection of a particular COTS product will have an appreciable effect. Stakeholder relevancy can be determined by the "hole" the products are trying to fill or by the constraints imposed by the products.

Evaluation stakeholders may not be a proper subset of the stakeholders who are identified for the system, since the scope of the expectations for a COTS product and vendor are sometimes different than documented expectations for the system.

As additional stakeholders are identified, some may become members of the evaluation team. However, the size of the team normally must be limited to avoid situations of broad participation with no progress. Practical experience suggests that the core working team should be limited to approximately 7-8 individuals. If there are a larger number of stakeholders, multiple sessions and management of various groups may be necessary.

Picking the approach. Next, planning determines the basic characteristics of the evaluation activity. Some of the parameters of the approach selected include the depth or rigor of the evaluation, the basic strategy for selection, and the number of iterations ("filters") needed to reduce the number of candidate products.

Some evaluations must be extremely rigorous while others are successfully accomplished with far less rigor. More rigorous evaluations that yield more accurate results will be used in cases where the system risks from failed products are high, while less rigorous techniques will be used where the risk from failed products is lower. Two factors that determine the depth or rigor of an evaluation are:

- the likelihood the wrong product will be selected, given a specific level of rigor of the evaluation
- the potential impact or system risk incurred if the wrong selection is made

To identify the necessary depth of a PECA evaluation, the criticality of the component and the candidate products should be considered. There is a spectrum of criticality according to which one can select appropriate approaches to evaluation.

For situations involving low technical risk and low involvement with strategic objectives, less evaluation effort and precision is required. For the lowest possible risk, a near-random selection (pitching pennies into fishbowls) may be justifiable.

For situations involving moderate technical risk or that have a significant, but not all-pervasive impact on the strategic objectives, moderate effort and precision of evaluation are required. The evaluation can focus on the specific discriminators between the various products that indicate some useful enhanced capability.

For situations involving high technical challenge or risk or are critical strategic objectives, the greatest effort and precision is required. These situations normally involve the potential for great financial, environmental, or property damage, or the harming or loss of life. This class of COTS implementation justifies the greatest rigor in COTS evaluation. In most situations, the best approach is to employ a methodic research process to gather necessary data.

A selection strategy involves the basic algorithm that will be used to identify an appropriate product. Two common selection strategies are used: first fit and best fit. First fit can be used when the selected product must fill a well-understood core set of needs. In this case, additional "goodness" of a product is unimportant or it is not worth extra evaluation costs. First fit considers minimum requirements and answers the question, "Is it good enough"? This does not imply that the set of criteria by which products are assessed is any less stringent or complete. The only implication is that the first candidate found that meets all requirements is selected without comparison to other candidates' capabilities.

Best fit is used when there is an appreciable gain in getting more than the minimal amount of some characteristic, or when no candidates are likely to meet all requirements. For example, in some situations a minimum performance is specified, but better performance adds significant value to a product within the context of the system. Best fit answers the question, "How good is each product"?

Sometimes it is not reasonable to evaluate all candidates because the number is too large. When this is the case, there must be a way to reduce the number of candidates that are considered for in-depth (and more costly) evaluation. The solution is to develop one or more "filters," which are inexpensive ways of eliminating candidates. Factors to consider in deciding whether to use filters and how many to use include the size of the field of candidates, the availability of discriminating criteria, and the evaluation budget. Each filter by itself may represent a full iteration through the PECA process (i.e., careful planning, establishment of criteria, etc.), or it may be more appropriate to include multiple filters in a single PECA process iteration.

3.1.5 Estimating Resources & Schedule

Unfortunately, there are few specific techniques available for estimating resources and schedule for COTS evaluation. COCOTS [3] is one of the few attempts to address the costs associated with building a COTS-based system. However, the technique does not isolate the costs associated with COTS evaluation.

Fortunately, general techniques with which you are already familiar are applicable, such as expert opinion, analogy, decomposition, and cost modeling. Some of the COTS-specific factors that may affect your estimates include:

- The level of rigor required. In general, the more rigorous the evaluation, the greater the short term cost. However, rigorous evaluations may lower long-term costs in building the system by avoiding the wrong choice.
- The number of candidates being evaluated: The more candidates evaluated, the higher the overall cost.

- Your evaluators' experience and availability: Evaluation costs are often higher when evaluations are performed by experienced evaluators, as they tend to perform more rigorous evaluations. However, use of experienced evaluators can be expected to reduce costs down the road.

We have seen cases in which inadequate resources are allocated to critical COTS evaluations and other cases where excessive time and effort are spent for trivial ones. It is important that the effort expended match the importance of the product decision.

3.2 Establishing Evaluation Criteria

Evaluation criteria are the facts or standards by which the fitness of products is judged. Evaluation criteria should be derived from requirements. As noted previously, however, system requirements rarely address the specific concerns that determine whether a COTS product is viable in a particular setting. Thus, the first step in establishing evaluation criteria must be determining appropriate evaluation requirements. Evaluation criteria are then constructed from these evaluation requirements.

Identifying evaluation requirements. There are actually two problems associated with identifying evaluation requirements. The evaluation team must determine which system requirements are legitimate requirements for the COTS product, and the team must determine any additional evaluation requirements that are not directly derived from system requirements.

Normally, a single COTS product is not expected to satisfy every system requirement. Therefore, the subset of system requirements that are applicable to the COTS products under consideration must be identified. This activity is called applicability analysis. Since COTS products are not mirror images of each other, it often occurs that different candidates will fulfill different subsets of system requirements.

Even after system requirements are analyzed for applicability, there are likely to be additional requirements on the COTS product that are not yet documented. Examples of legitimate evaluation requirements that are not always addressed by system requirements include:

- Architecture/Interface constraints - COTS product decisions are often constrained by other decisions that have already been made. These constraints become evaluation requirements. For example, if a decision has been made to use CORBA [4] as the middleware mechanism, it makes little sense to select a product that conflicts with this technology.
- Programmatic Constraints - Time, money, available expertise, and many other programmatic factors may be sources of evaluation requirements.
- Operational and Support Environment - Not all aspects of the operational and support environment are included as system requirements. For example, information about the organization that will perform maintenance on the system is frequently omitted.

Regardless of whether evaluation requirements are derived from system requirements or from additional expectations placed on COTS products, errors can

arise. Some errors arise from assigning too many requirements to a particular evaluation. This can result in the elimination of suitable COTS products because they don't meet all of the requirements. An example of this is the tendency to want every "cool" capability offered in the COTS marketplace. To combat this tendency, consider the risk to the system mission should the feature be absent.

Other errors occur when the set of evaluation requirements is incomplete. This reduces the scope of the evaluation and can result in the selection of unsuitable COTS products. Insufficient understanding and oversimplification of the problem can cause these errors. An iterative approach to building evaluation requirements and evaluating products will help mitigate this risk. As you gain understanding about the problem you will inevitably identify requirements that were initially overlooked.

Constructing Criteria. An evaluation criterion consists of two elements. These are a capability statement and a quantification method. The capability statement is a clearly measurable statement of capability to satisfy a need. The quantification method is a means for assessing and assigning a value to the product's level of compliance with the capability statement.

Well-defined criteria exhibit a number of common characteristics. First, they are discriminating, in that they allow the evaluator to distinguish between products. Criteria that are met by most or all products don't discriminate. For example, the presence of a graphical user interface will not (normally) discriminate between modern word processors. Including criteria of such limited value also dilutes the effort spent determining product performance on discriminating criteria.

Well-defined criteria also exhibit minimal overlap. If criteria overlap, then the associated product characteristics can be factored into deliberations multiple times, which can lead to wasted effort or misleading results. Finally, well-defined criteria reflect the context of the system that is being constructed. This calls into question the value of a list of products that was approved for use by some other organization, since the criteria used by that organization are unlikely to match those that are produced specifically for the system.

3.3 Collecting Data

Collecting data involves executing the evaluation plans to determine the performance of various products against the evaluation criteria that have been developed. However, the act of collecting data will often change the basic assumptions, since COTS software is full of surprises (a few good ones and more than a few bad ones). This is one of the reasons for applying an iterative approach to building COTS-based software systems – as the evaluator learns by collecting data, this new knowledge can be reflected in new concepts about the system and COTS products and new criteria for evaluation.

Different criteria and different situations require different data collection techniques. For example, the technique applied for determining the value of a critical criterion will likely be more rigorous than that applied for determining the value of a criterion that carries with it little risk.

The specific techniques you choose will be in part determined by the degree of confidence you need in your results. Obviously, the closer the technique comes to execution of the COTS component in your specific system context, the higher the degree of confidence you can have about how the product will perform in your actual system. Different families of techniques include:

- literature review - a wide variety of techniques with the common characteristic of being based on reviewing documents. Documents include user manuals, release notes, web based reports, product history, third party evaluations, etc.
- vendor appraisals - techniques that focus on the characteristics of the vendor that provides the product. Information about the vendor may be obtained from interviews, vendor literature, formal capability evaluations, independent financial analyses (e.g., Standard & Poor's), trade journals, and customer kudos and complaints (often published on web sites).
- hands-on experiments - techniques that employ and execute the actual COTS. Hands-on techniques are an essential part of a rigorous evaluation. They are necessary to verify vendor claims and to determine interactions with other components, the feasibility of proposed architectures and designs, and performance and reliability in the system context Hands-on techniques include product probes that investigate specific features of products, prototypes, scenario-based evaluations, benchmarks, experimental fielding, and product demonstrations in which users assume control.

Determining how a specific product (or products) stacks up against the criteria is not the only data that can be gathered while collecting data. In some situations it may not even be the most important result. For example, the improved understanding of the COTS marketplace and of the system context gained during COTS evaluation is an invaluable contribution to the development of the system. Some of the many less obvious results that should be captured during data collection include the degree of confidence in data, the system architecture and design implications of the selected product, limitations and conditions on how the product is used, and deficiencies in assessment methods, evaluation requirements or criteria.

3.4 Analyzing Results

Data collection typically produces a large number of data, facts, and checklists. This raw data must be consolidated into information that can be analyzed. Consolidation does not compare products; it simply makes sense of data. Analysis is required for reasoning about the data collected.

Data Consolidation. Consolidation almost always implies some loss of detailed information. This is the price that is paid for condensing a large mass of information into some more quickly comprehensible format. A balance must be struck between the need for easy understanding (a high level of consolidation) and the risk of losing too much information. For example, weighted aggregation [5] is commonly used to condense values for all criteria into a single overall fitness score. This technique provides a quick, but often misleading comparison of products since high levels of

consolidation can make two very different products can appear to be virtually identical.

Data Analysis. Data analysis involves reasoning about the consolidated data in order to make a recommendation. Analysis is a very creative task, and the best approach is simply the application of sound and careful reasoning. There are, however, three particularly useful techniques: sensitivity analysis, gap analysis and analysis of the cost of repair.

Gap Analysis highlights the gap between the capability provided by a COTS component and that capability required for the system. A gap analysis typically uses a matrix of product performance against evaluation criteria, where the individual cells contain information about how well a product fulfills the criterion, or a description of what functionality is lacking.

Sensitivity Analysis considers how the evaluation results react to changes in assumptions – for example, changes in the weighting of criteria or scoring by judges. By evaluating the sensitivity to changes in assumptions, it is possible to determine the impact of slight changes in judgments on recommendations of products.

Cost of repair [6] assumes that the evaluated products do not fully meet the system needs. Analysis of the cost of repair focuses on the implications to the system if a product is selected by considering the work that must be done to the system to repair deficits in the product. Deficit does not necessarily refer to a product flaw, but to a capability that is required in the system that the product does not demonstrate. Deficits may be repaired in many ways (e.g., by altering system architecture, adding additional functions, or modifying the requirements). Also keep in mind that a deficit may be caused by an overabundance of features as well as a paucity of features. "Cost" is not necessarily in terms of dollars; it could be in time (delays), shifted risks, etc.

3.4.3 Making Recommendations

The goal of evaluation is to provide information to the decision-maker. The evaluators must focus their recommendations on the information that the decision-maker needs. This can vary according to the type of organization and the characteristics of the decision-maker. For example, the decision maker at a bank emphasized that the evaluation demonstrate due diligence, such that any decision could be justified to bank investors. This emphasis "flavored" both the type of data gathered and the format and content of recommendations presented to the decision maker.

There are three main outputs of the PECA process:

- The *product dossier* is a repository of software documentation, discovered facts, assessment results, classifications, etc. that details all that is known about a given product at a point in time. There is one product dossier for each product evaluated. For a product that is selected, the product dossier serves as a source of information for the team that will architect, design, and integrate the system using the product
- The *evaluation record* is a description of the evaluation itself. Information in the evaluation record includes evaluation plans; personnel involved; dates and details of meetings and evaluation tasks; environment or context in which the products

were evaluated; specifics about product versions, configurations, and customizations; results all evaluation activities; rationale for decisions made; and lessons learned that might be useful for subsequent evaluations.

- The *Summary/Recommendations* document provides a synopsis of the evaluation activity and the resulting findings, along with the message the evaluation team wants to convey to the decision-maker. It includes both the team's analysis of fitness and of evaluation deficiencies (e.g., any need for further evaluation, confidence in results.

4 Conclusions

Some individuals believe that following any documented process is a waste of time and money, particularly when the end goal is to save time and money (as it often is with a COTS solution). Our experience in analyzing troubled programs is that all too often highly informal COTS evaluation processes share the blame for the failure. But the process described here is a *means* of performing COTS evaluations and not an end in itself. Expect to tailor this process for your own situation, and do not let it get in the way of getting good data and making an informed recommendation.

Regardless of the COTS evaluation process you adopt, remember that COTS evaluation is an ongoing activity. Your organization will need to evaluate new product versions and potentially identify product replacements over the life of your system. If you have a foundation of good evaluation processes and practices, along with good documentation of the characteristics of products and the rationale for decisions, you have a good start at making COTS products work for you.

References

1. International Organization for Standardization (ISO). ISO/IEC 14598-1:1999 – Information Technology -- Software Product Evaluation. (Geneva, Switzerland). 1999.
2. Briand, L., Morasca, S., and Basili, V.R. *Goal-Driven Definition of Product Metrics Based on Properties* , CS-TR-3346, UMIACS-TR-94-106, University of Maryland, Computer Science Technical Report, December 1994, 24 pages. More information available online at: http://sel.gsfc.nasa.gov/website/exp-factory/gqm.htm.
3. Abts, Christopher M. and Boehm, Barry. *COCOTS (Constructive COTS) Software Integration Cost Model: An Overview,* USC Center for Software Engineering, University of Southern California, briefing. Available online at: http://sunset.usc.edu/COCOTS/cocots.html (August 1998).
4. *The Common Object Request Broker: Architecture and Specification*, Version 2.0. Framingham, MA: Object Management Group, 1996. Also available online at URL: http://www.omg.org.
5. Saaty, T., The Analytic Hierarchy Process, McGraw-Hill (New York, 1990).
6. Wallnau, K.; Hissam, S.; and Seacord, R. *Building Systems from Commercial Components.* Addison Wesley 2001. ISBN: 0201700646.