

**Guidelines
For
Producing COTS Assessment Background,
Process, and Report Documents**

**COTS Assessment Background (CAB)
COTS Assessment Process (CAP)
COTS Assessment Report (CAR)**

**USC-CSE
April, 2004**

Version History

Date	Author	Version	Changes made
3/31/03	Barry Boehm	0.1	<ul style="list-style-type: none">• Added outlines for CAB, CAP, CAR
8/3/03	Ye Yang	0.2	<ul style="list-style-type: none">• Elaborated CAB, CAP, CAR
1/	Ye Yang	0.3	<ul style="list-style-type: none">• Modified part of CAB
2/3/04	Ye Yang		<ul style="list-style-type: none">• Added some details of CAB
3/1/04	Ye Yang		<ul style="list-style-type: none">• Added some details of CAP, CAR
3/5/04	Ye Yang		<ul style="list-style-type: none">• Added some details of CAP

1. Introduction

The Guidelines for COTS/NDI Assessment-intensive project will be introduced in this document. Please read the following general guidelines carefully, before proceeding to the guidelines for the individual deliverables.

1.1 Scope

The Guidelines for Producing COTS Assessment Background, Process, and Report Documents (CAB, CAP, and CAR) apply to projects that need to assess the relative merits of commercial-off-the-shelf (COTS), non-developmental-item (NDI), and/or other pre-existing software products/components for use in a software system.

Basically, COTS/NDI assessment activity takes place in the following two primary situations:

- As part of the software process for a COTS-based development project that follows general guidelines such as Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), Model-Based Architecting and Software Engineering (MBASE), Rational Unified Process (RUP), or Team Software Process (TSP);
- As a standalone COTS/NDI assessment activity to serve as the basis for future project decisions.

The guidelines cover the above two primary situations and were developed by integrating COTS-based development lessons learned and proven engineering practices to help teams prepare, plan, manage, track, and improve their COTS assessment activity.

1.2 Definitions

Commercial-Off-The-Shelf (COTS). We adopt the SEI COTS-Based System Initiative's definition [1] of a COTS product: 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;
- Used without source code modification.

Non-Developmental Item (NDI). NDI's are used in a similar manner to COTS products, but with one or more of the COTS criteria handled differently. Examples are:

- Open source software: Has COTS product attributes a, c, and d; but not b, and optionally not e (making your own changes to open source software means that you have to integrate these changes with all future open source product releases).
- Customer-furnished or corporate-furnished software: has COTS product attributes c and d, but not a and b, and optionally not e.
- Reuse-repository software: can assume many forms, with practically any combination of COTS product attributes a, b, c, d, and e.

COTS-Based System (CBS). CBS is generally defined as "any system, which includes one or more COTS products." This includes most current systems, including many which treat a COTS operating system and other utilities as a relatively stable platform on which to build applications. Such systems can be considered "COTS-based systems," as most of their executing instructions come from COTS products, but COTS considerations do not affect the development process very much.

COTS-Based Application (CBA). To provide a focus on the types of applications for which COTS considerations do affect the development process, we define a *COTS-Based Application* [3] as a system for which at least 30% of the end-user functionality (in terms of functional elements: inputs, outputs, queries, external interfaces, internal files) is provided by COTS products, and at least 10 % of the development effort is devoted to COTS considerations. The numbers 30% and 10% are not sacred quantities, but approximate behavioral CBA boundaries observed in historical CS577 projects. Obviously, CBA projects are only a subset of CBS projects.

COTS Activity. In our seven years of iteratively defining, developing, gathering project data for, and calibrating the COCOTS cost estimation model, we have identified three primary sources of project effort due to CBA development considerations. These are defined in COCOTS as follows:

- COTS Assessment is the activity whereby COTS products are evaluated and selected as viable components for a user application.
- COTS Tailoring is the activity whereby COTS software products are configured for use in a specific context. This definition is similar to the SEI definition of “tailoring” [4].
- COTS Glue Code development and integration is the activity whereby code is designed, developed, and used to ensure that COTS products satisfactorily interoperate in support of the user application.

2. Context

COTS and NDI are similar in that their use can significantly reduce software development costs and schedules by avoiding the need to develop their capabilities. But using several COTS/NDI components involves a number of significant risks, including problems with incompatibilities, controllability, release synchronization, and life cycle support.

Using COTS/NDI also has significant impacts on software processes, especially those of COTS-Based Applications. For example, unlike traditional type of application development, having specific and frozen requirements set in the early stage is risky for COTS/NDI based development because COTS/NDI may introduce more changes later on. These changes stem from mismatch between COTS/NDI and the desired system capabilities discovered afterwards, new releases of COTS/NDI products, newly introduced COTS/NDI products, COTS/NDI market sectors changes, or application infrastructure changes.

The choice of COTS/NDI components will often change a system’s originally-defined requirements and architecture by making alternative COTS/NDI options more cost-effective. This establishes the need addressed here for a COTS/NDI assessment process to determine the right combination of COTS/NDI components, requirements, and architecture for the system.

Most COTS/NDI assessments will be carried out in the context of software system development. In this case, the contents of the three key COTS/NDI guideline documents can be simplified by integrating them with the contents of whatever development guidelines are being used for development (e.g. DSDM, FDD, MBASE, RUP, TSP). But in some cases, a customer will want only a standalone COTS/NDI assessment, to serve as a basis for future development decisions. These Guidelines are written to cover this self-contained case.

COTS/NDI assessments are also carried out in an overall process context that also includes COTS/NDI tailoring, glue code development and integration. This process context is described with examples in [3] and the detailed process guidelines for COTS tailoring, and COTS glue code is under development and will be available soon in separate documents.

3. Key Documents

The key documents involved in COTS/NDI assessment are the COTS/NDI Assessment Background (CAB), COTS/NDI Assessment Process (CAP), and COTS/NDI Assessment Report (CAR). They are defined in a minimum-essential, tailoring-up fashion rather than in an exhaustive, tailoring-down fashion. As an MBASE example, the minimum-essential content of the CAB is Sections 2 (Shared Vision), 4.3 (Capabilities), and Section 4.4 (Levels of Service) in the Operational Concept Description (OCD). These provide the minimum-essential set of objectives, constraints, priorities, and situation background needed to perform the COTS/NDI assessment. All of the rest of the MBASE OCD, SSRD (requirements), and SSAD (architecture) documentation is optional, to be added only when the risks of not documenting something outweigh the COTS assessment costs and risks of documenting it.

The CAB, CAP, and CAR are living documents. They should be lightweight and updated whenever new risks, opportunities, or changes emerge.

COTS/NDI Assessment Background (CAB) Document

Preface

The CAB document is to describe the organizational needs and constraints, as well as conditions of the related commercial market sections. It should provide the minimum essential set of organization objectives, constraints, priorities (OC&P's), and situation background needed to perform a COTS/NDI assessment. It can be tailored up on a risk-driven basis to cover special COTS/NDI assessment concerns.

1. Introduction

1.1 Purpose and Scope

This section summarizes the purpose of the CAB document, the scope of the COTS assessment, and identifies the project stakeholders and possible COTS candidates. This should include

- “The purpose of the document is to provide the minimum essential set of objectives, constraints, priorities, and situation background needed to perform a COTS/NDI assessment for the [name of system].
- Its scope covers the COTS/NDI assessment aspects of the [name portions] portions of the system plus the additional complementary activities of [name activities*].
- (Optional) Current life cycle phase and milestone.
- The description of project stakeholders, including their names, organizations, titles and roles in the COTS assessment.
- The list of current COTS candidates.”

* Example complementary activity: market trend analyses and product line analyses.

1.2 Reference

Provide a complete set of citations to all sources used or referenced in the preparation of this document. The citations should be in sufficient detail that the information used in this document can be traced to its source. Sources typically include books, papers, meeting notes, tools, and tool results.

1.3 Change Summary

For each version of the CAB document, describe the main changes since the previous version and briefly explain why. The goal is to help a reviewer focus on the most critical parts of the document needing review. The following change sources must be included:

- OC&P's changes (OC&P's newly introduced, removed, relaxed, and/or reprioritized);
- COTS changes (COTS candidates added, discarded, and/or updated);
- COTS Vendor changes (new Vendor, new Vendor claim, or vendor support changes);
- Other changes.

2. Overall System Objectives, Constraints, and Priorities

For COTS intensive projects, overall system OC&P's work not only as the shared vision among all stakeholders, but also as the basis from which the COTS evaluation criteria and testing scenarios are established. Additionally, COTS-assessment intensive projects are usually holding a high degree of requirement flexibility for the sake of a better compromising and balancing between what available best COTS options can provide and what the clients want and can afford.

Hence, for COTS assessment intensive projects, though the details of the OC&P's may need to be modified at every assessment cycle, it is very important to keep the OC&P's clearly stated so that the COTS assessment can start with the essential OC&P's set and the stakeholders can negotiate and reexamine them when reviewing intermediate assessment results. This way, the OC&P's can get continuously refined as the assessment goes into further details.

2.1 System Objective Description (SO)

A concise description of the system objectives is presented here, focusing on critical system capabilities in terms of customer’s needs, and relating them to evaluation criteria. It should take the following form:

- For (target customer)
- Who (statement of the need or opportunity)
- The (project name) is to investigate the feasibility of using (COTS/NDI categories) in the (product name)
- That (statement of key benefit-that is, compelling reason to selecting and integrate the recommended solution)
- Unlike (developing from scratch or other comparative alternatives)
- Our product (statement of primary advantages resulted from COTS/NDI assessment)

Here is an example for a collaborative services system: “Our client from xxx organization need a stronger, user-friendly online collaborative services system to better support its faculty, staff, and students’ tremendous amount of daily collaborative work. Our project would search for some COTS collaborative packages, and perform thorough assessment on their fitness for xxx Collaborative Services System. Our final COTS recommendation to the client, xxx, unlike other same kind COTS products, ours should be mature, easy to integrate, and cost-effective for our client.

2.1.1 Benefits Realized

This section should briefly discuss the resulted organizational benefits in terms of future profitability, reputation, or market capitalization with the investment of the system to be built. Next section about “Results Chain” helps to explain these benefits realization path.

2.1.2 Results Chain

Results Chain starts from the initiative of performing COTS assessment rather than implementation of the system. Some assumptions and contributions related to COTS assessment activity may also be identified in the Result Chain diagram.

Figure 1 shows a simple Results Chain provided as an example for COTS assessment projects. It establishes a

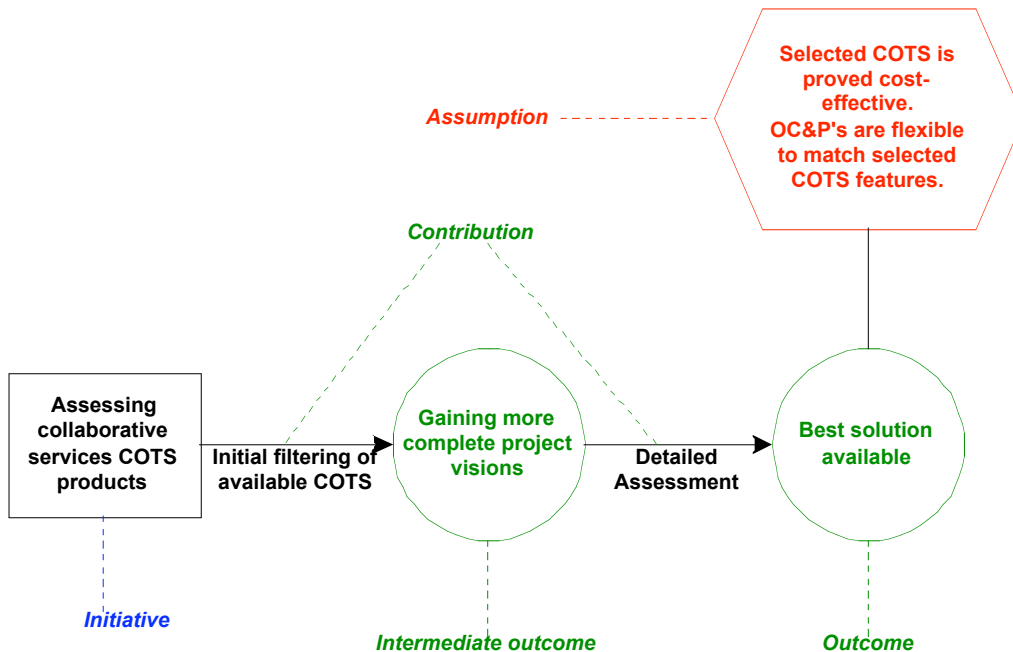


Figure 1 Benefits Realization Approach Results Chain

framework linking Initiatives that consume resources (e.g., performing COTS assessment on online collaborative services/COTS products) to Contributions (not delivered systems, but their effects on existing operations) and Outcomes, which may lead either to further contributions or to added value (e.g., increased savings due to higher productivity by integrating the best COTS solution). A particularly important contribution of Results Chain is the link to Assumptions, which condition the realization of the Outcomes.

The Result Chain provides a valuable framework by which your project can work with your clients to identify additional non-software initiatives that may be needed to realize the potential benefits enabled by the project initiative. These may also identify some additional success-critical stakeholders who need to be represented and “brought into” the shared vision.

2.2 Key Stakeholders

Identify each stakeholder by their home organization, their authorized representative for project activities, and their relation to the Results Chain. The four classic stakeholders are the software/IT system’s users, customers, developers, and maintainers. Specifically, for a COTS assessment-intensive project, key stakeholder should include the following parties:

1. Customer: Project initiator.
2. User: The people that are going to use the proposed system, it is recommended that the COTS assessment should involve the end user into the evaluation process, present evaluation results (intermediate & final results) to them, collect COTS usage information (if trial version is available) from them, and perform appropriate training for them to experiment COTS(s).
3. Domain experts: A new and important stakeholder role introduced for COTS assessment projects. These are the people from the customer organization that can provide feedbacks and advices for solving some domain-specific problems, such as helping in acquiring and prioritizing OC&P’s, or helping to reconcile conflicts between COTS product and desired OC&P’s during COTS assessment process.
4. COTS Experts: A new and important stakeholder role introduced for COTS assessment projects. These are the technical representative(s) from the COTS vendor side that can provide technical help, such as helping in answering questions during the process of obtaining, installing, experimenting with, and analyzing the test version of the COTS product. For example, technical from COTS vendors are recommended to involve into the evaluation process.
5. COTS vendors: A new and important stakeholder role introduced for COTS assessment projects. They provide inputs of COTS information (such as functionality, performance, price model, and possible future evolution direction, etc.) and test version of the product to be assessed.
6. Other necessary stakeholder like system administrator and maintainer, the definition and responsibility of them are similar to traditional development projects. Note that their specific responsibilities and activities may be highly COTS-dependent.
7. Evaluator: usually called “developer” in cs577 projects. They need to learn and master certain knowledge besides conventional software development scope, such as ability to explore information through market survey, to make regular contact and question & answering sessions with COTS vendor, domain experts, and COTS experts, to report and renegotiate with the customer according to the progress of COTS assessment, and to make final COTS recommendation to the customer.

Additional stakeholders may be system interfaces, subcontractors, suppliers, venture capitalists, independent testers, and the general public (where safety or information protection issues may be involved).

Common Pitfalls:

- Not paying much attention and effort in order to get the newly introduced COTS assessment stakeholders involved. Many of the delays in schedule for COTS intensive projects are due to the inability to hook with COTS vendors, domain experts, or COTS experts along the assessment process to collect as complete information about COTS as possible and regularly collect for updates as well.
- Being too pushy or not pushy enough in getting your immediate clients to involve the other success-critical stakeholders. Often, this involves fairly delicate negotiations among operational organizations. If things are going slowly and you are on a tight schedule, seek the help of your higher-level managers.

2.3 COTS Assessment Boundary and Environment

The system boundary distinguishes between the services your project will be responsible for assessing the fitness of the candidate COTS solution(s), and the stakeholder organizations, COTS vendors and interfacing systems for which your project has no authority or responsibility, but with which your project must coordinate to realize a successful COTS assessment and its resulting benefits.

Figure 2 shows the COTS assessment context diagram used to define the system boundary. It shows the type of information that may be included in COTS intensive project’s context diagram, but is not intended to be a one-size-fits-all template.

- The Context Diagram for the COTS assessment should include entities for all the key operational stakeholders described above (CAB 2.2)
- The “Top-level system objectives” box defines the boundary of desired system. It should include the list of top-level system capabilities that your COTS assessment will be based upon.

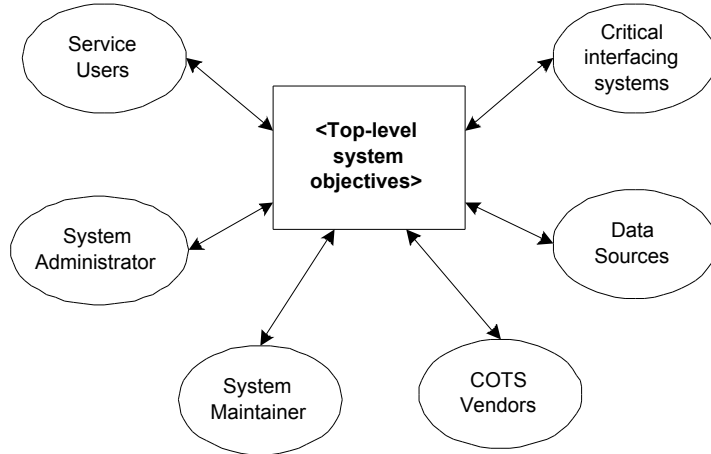


Figure 2 COTS Assessment Context Diagram

Common Pitfalls:

- Including system design details

2.4 Major Project Constraints

Describe any constraints that are critical to the success of COTS assessment, such as:

Table 2-1 Project Constraints Specification

Identifier:	<<Give a reference number and name>> such as “CST-1”
Name:	Limited Budget
Description:	<<Describe the constraint>>, such as “The customer organization is only willing to buy COTS within \$25k.”
Influence and on COTS assessment:	<<Indicate how this constraint can affect the COTS assessment>>, such as “Need to add an evaluation criteria to filter out those COTS candidates that costs more than \$25k”
Measurable:	<<Indicate how this constraint can be measured with respect the specific elements it addresses, or what needs to be looked at within the project to see that the constraint has been adhered to >> E.g., “Cost analysis can provide estimation cost along each solution”.
Relevant:	Relative to COTS Evaluation Criteria Table CEC X.X
Specific:	<<Describe what particular aspects of the constraint>> e.g., “\$25k”. (There is no need to repeat such information if it is obvious from the above information.)

3. Domain/Organization Description

This section should refine the shared vision discussed in section 2. It should provide sufficient detail so that the stakeholders will understand the rationale for the COTS assessment, and will be able to evaluate the goals of the COTS assessment, and the success factors for the COTS assessment.

This section should be written using terms understood by all the stakeholders in the project, especially customers and domain experts. Do not become so technical that customers and high-level managers cannot understand.

3.1 Organization Background

Provide a brief overview (a few sentences) of the organization(s) sponsoring the COTS assessment of the system, the organization that will be the users of the system, and the organization that will be maintaining the system. (These organizations may or may not be the same organization.)

Recommendations

- Consider using each group's mission statements and/or their objectives and goals as the basis for this section.
- Do not get carried away with details that are not centrally relevant to the new system.

3.2 Current Organization Environment

In the following sections, describe the current environment of the organization. (This model is sometimes called an *"as is" model*) The description should include the structure, artifacts, processes, and shortcomings of the organization's environment, which should be satisfied and addressed by one COTS solution obtained through the COTS assessment.

3.2.1 Structure

Briefly describe the current workers (e.g. people *roles*, systems) of the organization and the *outside actors* that interact with the organization (e.g. customers, suppliers, partners, prospects). Each worker and outside actor should be relevant to the current organization's processes (CAB 3.2.3). Identify which workers and outside actors interact with other workers and actors. Provide a brief description of its role, purpose, and responsibilities.

Recommendations:

1. Give each worker and outside actor classifier a name that expresses the responsibilities of its instances.
2. A good name is usually a noun (e.g. "librarian"), a short noun phrase (e.g. "department secretary"), or the noun form of a verb (e.g. "administrator").
3. Avoid names that sound alike or are spelled alike, as well as synonyms.
4. Clear, self-explanatory names may require several words.
5. To facilitate traceability, assign each worker or outside actor classifier a unique number (e.g. Worker-1, BActor-10).

3.2.2 Artifacts

Briefly describe the current *artifacts* (e.g. documents, products, resources) inspected, manipulated, or produced by the organization, and the relations among the artifacts. Each artifact identified should be relevant to the current organization's processes (CAB 3.2.3).

Your customer can give you information about the current artifacts.

- What are the major documents inspected, manipulated, or produced by the organization?
- What are the resources used in production by the organization?
- What are the items produced by the organization?
- What is its general purpose, role, or description for each artifact?

3.2.3 Processes

Describe the operational processes within the current organization used to fulfill its business goals. For each process, identify which workers and outside actors participate in the process, and which artifacts are inspected, manipulated, or produced by the process; and describe at a high level the actions performed by each worker and outside actor during the process.

A major objective of the process model is to provide a context for evaluation criteria and test cases of COTS assessment to be developed in CAR section 4. For example, for a process of "the proposed system will eliminate or make efficient manual order entry and verification steps" described here, in CAR section 4, there should be a related evaluation criteria or business test scenario to this process.

Representation:

Create a list of process names and a story to describe each process. Each story is a paragraph describing “something the system should do” (used in Xtreme Programming [6] and some Agile Methods).

Table 1-1: Business Use-Case Description

Identifier:	Unique identifier for traceability (e.g. Process-xx)
Use-Case Name:	Name of use-case
Purpose:	Brief description of purpose
Priority:	Relative importance of the process to the business
Flexibility:	Must-have or nice-to-have
Worker or outside actor:	List of worker or outside actors participating in the use-case
Pre-conditions:	Description of state that workers, outside actors, and artifacts should be in before use-case performed. (informal text, OCL1, or both)
Post-conditions:	Description of state that workers, outside actors, and artifacts are in after use-case performed. (informal text, OCL, or both)
Actions Performed:	Describe actions performed by a worker or an outside actor.

Recommendations:

1. Describe only those processes that are relevant to the proposed system (e.g., processes that the proposed system will participate in).
2. Give each process a name that expresses the behavior that it represents.
3. A good name is usually a verb, or verb phrase.
4. Each name must be unique relative to the containing package. (Two classifiers in different packages can have the same name.)
5. Avoid names that sound alike or are spelled alike, as well as synonyms.
6. Clear, self-explanatory names may require several words.
7. To facilitate traceability, assign each process a unique number (e.g. Process-01 or BUC-01).
8. Avoid overly technical- or implementation-related processes unless they are already present in the current system. An example of an appropriate level process for an Order Entry System would be Add New Sales Item To Order.

Common Pitfalls:

- Describing a worker or customer in a process description and not describing the worker or customer in the structure of the current organization (CAB 3.2.1).
- Describing an artifact in a process description and not describing it in the artifacts of the current organization (CAB 3.2.2).
- Including processes that are not included in the Prioritized System Capabilities (CAB 4), and are not listed as evaluation criteria or COTS test case in (CAR 4).
- Including design details about the process. For example, specifying exactly how user validation will be performed.

3.2.4 Shortcomings

Describe limitations of the current organization environment and current system, if it exists. Focus on how the organization environment and current system needs to be improved, or replaced by a COTS-based solution, which is to be discovered through the COTS assessment effort.

¹ UML’s formal specification language, called an Object Constraint Language (OCL).

Recommendations:

1. Clearly and concisely describe each shortcoming.
2. To facilitate traceability, assign each shortcoming a unique number (e.g. Shortcoming-01).

4. Prioritized System Capabilities

Capabilities define broad categories of system behaviors, should realize the high-level system objectives described in COTS Assessment Boundary and Environment (CAB 2.3).

- Since the Proposed system is greatly COTS-dependent, it’s better not to go into too detailed or be too specific when describe system capabilities, entities, or activities.
- Bear in mind that keep the system capabilities flexible, COTS-driven when thinking and documenting.
- For the proposed system, project goals, constraints, capabilities, and level of services are first identified during the win-win session between customer and evaluators.
- Some of system capabilities produced through win-win negotiation might be proved wrong, biased or unachievable along the progress of the project, given the great uncertainties within COTS candidates will be detected gradually. In this case, introducing more sessions of win-win negotiations among key stakeholders to refine your system capabilities will be a recommendable approach.
- Describe a few capabilities and work with domain experts, and operational stakeholders, to clarify and refine them. As more capabilities are documented, architects get a better idea of how those people view the proposed system (I.e. the conceptual system from their perspective).
- Minimum information for each system capability is as indicated in the following suggested template:

Table 4-1 Prioritized System Capability Specification

Identifier:	Unique identifier (e.g. PSC-x)
Name:	Name of capability
Description:	What the capability allows the operational stakeholders to do
Importance:	Relative importance of the capability: (e.g. 1..n or Primary Secondary Optional)
Flexibility:	Is this capability: must-have or nice-to-have?
Used In	Reference to organization processes (CAB 3.2.3) that use the capability.

Common Pitfalls:

- Including a system requirement
- Including a system Levels of Service goal
- Including a system behavior
- Including a lot of capabilities for a small system (some of them are likely to be either system requirements or system behaviors)

5. Desired and Acceptable Levels of Service

- Define levels of service required in the System (i.e., “how should” and “how well” the system performs a given capability).
- Indicate how the Desired and Acceptable Levels of Service are relevant to the Objectives, Constraints, and Priorities.
- Levels of Service correspond with Spiral Model Objectives or in some cases constraints, as when the level is a non-negotiable legal requirement.
- It is recommended to specify both acceptable and desired quality levels, and leave the goals flexible to produce the best balance among Level of Service Requirements (since some Level of Service Requirements conflict with each other, e.g., performance and fault-tolerance) among different COTS solutions.
- Levels of Service should be M.R.S. (Measurable, Relevant, Specific). Measures should specify the unit of measurement and the conditions in which the measurement should be taken (e.g., normal operations vs. peak-

load response time). Where appropriate, include both desired and acceptable levels. Again, don't get too hung up on measurability details.

Table 5-1 Levels of Service Specification

Level of Service:	Give a reference number and name, such as “LS-1: Response time”
Description:	Describe the level of service, such as “1 second desired; 2 seconds acceptable”
Degree of Flexibility:	Desired or acceptable
Measurable:	Indicate how this goal can be measured with respect the specific elements it addresses – include as appropriate baseline measurements, minimum values, maximum values, average or typical or expected values, etc., such as “time between hitting Enter and getting useful information on the screen”
Relevant:	Describe which system capabilities (CAB 4) this level of service is relevant to, such as “larger delays in order processing (see capability 3 in CAB 4) cause user frustration”
Specific:	Describe what in particular within the system capabilities (CAB 4) this level of service addresses, such as “credit card validation may cause significant delay when attempting to connect to the verification service”

Common Pitfalls:

- Overburdening the system with Levels of Service that are not validated by the customer
- Including superfluous Level of Service goals. Table 2 shows typical stakeholder concerns for Level of Service.
- Levels not satisfying the M.R.S. criteria

Table 5-2: Stakeholder Roles / Level of Service Concerns Relationship

Stakeholder	Roles and Primary Responsibilities	Level of Service Concerns	
		Primary	Secondary
General Public	Avoid adverse system side effects: safety, security, privacy.	Dependability	Evolvability & Portability
Operator	Avoid current and future interface problems between system and interoperating system	Interoperability, Evolvability & Portability	Dependability, Performance
User	Execute cost-effective operational missions	Dependability, Interoperability, Usability, Performance, Evolvability & Portability	Development Schedule
Maintainer	Avoid low utility due to obsolescence; Cost-effective product support after development	Evolvability & Portability	Dependability
Developer	Avoid non-verifiable, inflexible, non-reusable product; Avoid the delay of product delivery and cost overrun.	Evolvability & Portability, Development Cost & Schedule, Reusability	Dependability, Interoperability, Usability, Performance
Customer	Avoid overrun budget and schedule; Avoid low utilization of the system	Development Cost & Schedule, Performance, Evolvability & Portability, Reusability	Dependability, Interoperability, Usability

COTS/NDI Assessment Process (CAP) Document

Preface

The CAP document is organized in the same form used in MBASE plans. It covers the minimum essential “why/whereas, what/when, who/where, how, and how much” aspects of the activity being planned. It can be tailored up on a risk-driven basis to cover special COTS/NDI assessment concerns. However, it generally does not need detailed work breakdown structures, monitoring and control plans, configuration management plans, iteration plans, or quality management plans. The CAP is a living document, and should be updated as new risks, opportunities, or changes emerge.

1.Introduction

1.1 Purpose, Scope, and Assumptions (why, whereas)

The purpose of this document is to provide the minimum essential set of plans needed to perform a COTS/NDI assessment for the [name of system].

Its scope covers the COTS/NDI assessment aspects of the [name portions] portions of the system plus the additional complementary activities of [name activities].

The following assumptions must continue to be valid in order to implement the plans below within the resources specified:

- [list assumptions]

1.2 Reference

Provide a complete set of citations to all sources used or referenced in the preparation of this document. The citations should be in sufficient detail that the information used in this document can be traced to its source. Sources typically include books, papers, meeting notes, tools, and tool results.

1.3 Change Summary

For each version of the CAP document, describe the main changes since the previous version and briefly explain why. The goal is to help a reviewer focus on the most critical parts of the document needing review. The following change sources must be included:

- Activities changes;
- Schedule changes;
- Resource changes.

2. Milestones and Products (what; when)

This section describes the COTS/NDI assessment process for the [name system] system and the rational behind such process. It should contain schedules and milestones that indicate when each function and product will be completed.

2.1 Overall Strategy

Summarize and rationalize the overall COTS/NDI assessment strategy, including for each class of product being assessed:

- Specify the number and length of down selection iterations;
- Specify the list of COTS attributes to be used as evaluation criteria, ordered by the importance of each attribute; and
- Specify the types of assessment activities to be performed, example activities including:
 - Reference check
 - Document review
 - Supplier inquiry
 - Demo
 - Analysis
 - Execution test
 - Prototype application

Evaluation criteria and weights are established based on stakeholder-negotiated OC&P's for the system. Stakeholders also agree on the business scenarios to be used for the assessment. For a checklist of the strategy elements mentioned above, such as COTS classes, COTS attributes and their definitions, please refer to Appendix A "COCOTS Assessment component".

Besides, identify complementary activities and how they synchronize with the COTS/NDI assessment. Examples are market trend analyses and product line analyses.

2.2 Milestones and Schedules

Identify and sequence the COTS/NDI assessment planning, preparation, execution, and analysis activities, and show how they interact with external events involved in complementary assessment activities and overall milestone schedules. Prepare and keep up-to-date a Gantt chart and PERT chart of the activities and schedules, using Microsoft Project or the equivalent. Identify critical path dependencies (e.g., interoperability testing requires executable products, facilities, equipment, probably tailoring and glue code, test drivers, key personnel, etc.) as candidate risk items for Section 4.2.

An elaborated risk-driven Win Win Spiral model shown in figure 3-1 highlights the critical activities of a COTS assessment process that should be scheduled.

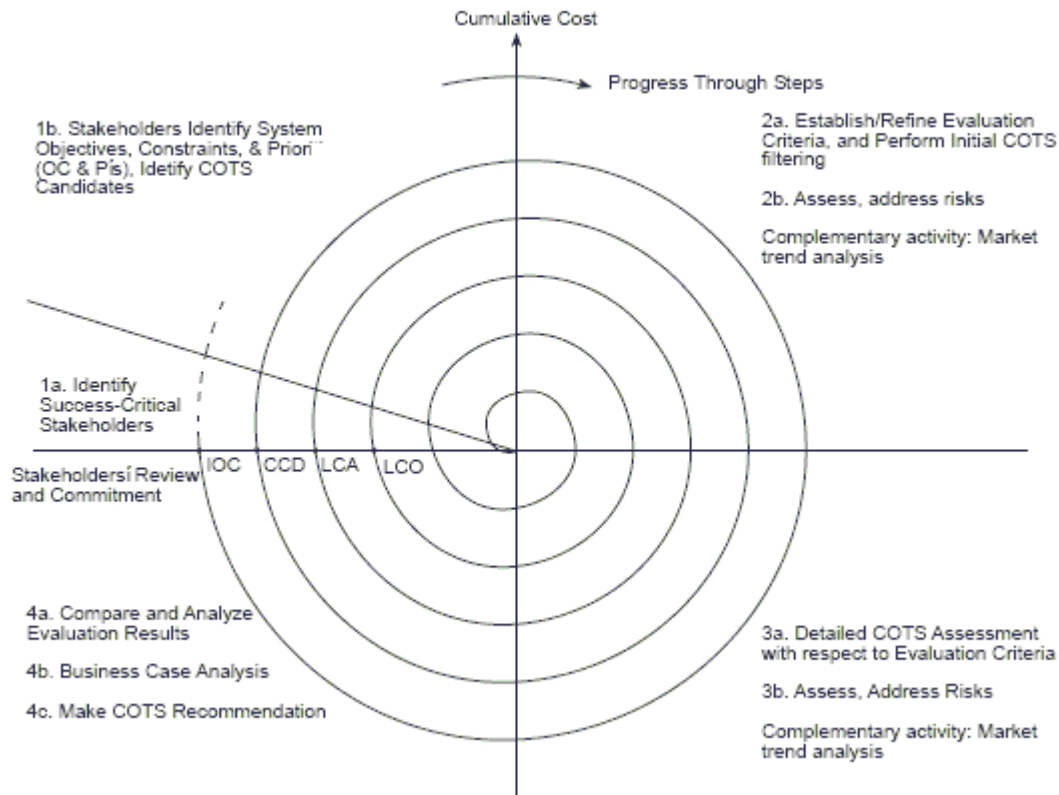


Figure 3-1 Elaborated Win Win Spiral Model for COTS Assessment Project

The following is an example of the milestone and schedule description for a spiral cycle of a COTS assessment project (Note that blue lines show new and important tasks):

Inception (2) Phase (Beginning of spiral cycle # 1 for 577b)

At this iteration it is required to map our plans to the win win spiral in a more detailed way. In addition to this it is required to define more detailed reviews where includes exit criteria with the quality management plan.

Identifying Stakeholders and system OC&P

- 1/26/03 [Checking new stakeholders of the project](#)
- 1/27/03 Team Formation
- 2/1/03 Review of MTA Criteria
- 2/2/03 Review of COTS Evaluation Criteria
- 2/4/03 Client Feedback
- 2/5/03 Feedback From Professors
- 2/6/03 [Defining Exit Criteria](#)
- 2/6/03 [Use Spiral Modeler for maintaining the Spirals](#)
- 2/10 RLCA draft on the Web

Evaluating Alternatives with respect to OC&P and

Elaborating Product (These 2 steps are done parallel)

- 2/12 [Address Possible Risks of the Project](#)
- 2/16 MTA Template changes
- 2/16 Continuing COTS evaluation for HP requirements (Oracle, Stelelent)

Verification and Validation of the product

- 2/20 RLCO (RLCA) ARB
- 2/23 [Peer Review of MTA and Evaluated COTS2/24](#) [IV&V Review](#)

2.3 Deliverables

Identify the content, completion criteria, and due dates of all deliverables. These could include interim and trial versions of the COTS/NDI Assessment Report (CAR), or reports summarizing market trend analyses or product line analyses. In some cases, customers may want reasonably documented versions of your evaluation software for continuing assessment purposes.

3. Responsibilities (who, where)

Identify key stakeholder roles and individuals, and summarize their responsibilities in planning, preparing, executing, analyzing, or reviewing the COTS/NDI assessment and related activities. Summarize the objectives, content, and stakeholder roles involved in key milestone reviews.

Stakeholders	Stakeholders' responsibility	When	Where
Evaluator	Creating, modifying, using, and reporting evaluation criteria&results to the client	win win negotiation and client meeting;	Team meeting and client meeting
	Research into COTS market sectors; consult with COTS vendors or business domain experts; compare available features of each COTS package; analyze and provide a reprioritized list of features for the proposed system	In pallel with the evaluation process, provide information for helping decision making in evaluation reviews	One team member should be assigned responsible for this task, reporting for team leader and other stakeholders
Client	Facilitate evaluator's creating, refining of template, review templates	Client meeting	Provide feedback about the templates
	Facilitate the research, the contract issues of COTS candidates, review analysis results	Client meeting, ARB	
COTS Vendor	Providing COTS documentation, price model, possible future release info, and helping to identify difficulty of meeting requirements gap by tailoring and glue code development	Any time as needed and regularly	Contact by email, phone, meeting

Domain Expert	Help in prioritizing requirements according to COTS provided features, and identifying the missing requirements from COTS	Any time as needed and regularly	Contact by email, phone, meeting
---------------	---	----------------------------------	----------------------------------

For the assessment performers, identify their specific responsibilities for types COTS/NDI products assessed, aspects assessed, types of assessment, and associated support activities.

4. Approach (how)

4.1 Assessment Framework

4.1.1 Instruments

Provide a description of the assessment instruments (manual forms, Web forms, spreadsheets, etc.) to be used in performing the assessments. For example, the following table shows the description of the instruments used in a COTS assessment project:

Assessment Instrument	Description
Evaluation Criteria and Weights	A Spreadsheet file that is created from system OC&P, 's and maintained by the 577 team.
COTS Product Literature	Product manuals obtained from vendor website/email/mail contact.
COTS Demos	Available from the vendor website/email/mail contact.
Evaluation Data Collection Form	A Spreadsheet file that is created from evaluation criteria, and used by the 577 team to fill in evaluation data for each COTS candidate.

4.1.2 Facilities

Identify the facilities, equipment, software, COTS licenses, and procedures involved in each type of assessment, particularly execution tests and prototype applications.

For example, provide a description of facilities required as table below:

Facilities	Description
Hardware	Computer(s) with access to the internet.
Software	Describe any software required in order to set up the COTS assessment environment.
COTS license	Describe the license type/status of each COTS candidate, and address possible risks (e.g. schedule delay due to unable to get COTS license) related to COTS license issue in CAP section 4.2.
Procedures	If execution test is performed, prepare section 4 of CAR the test procedures with respect to the Processes discussed in Section 3.2.3 of CAB.

4.1.3. COTS Assessment

Identify the approaches of COTS assessment performed in your project. For each of the approaches, identify the facilities, equipment, software, COTS licenses, and procedures involved in each type of assessment, particularly execution tests and prototype applications.

The following figure 3-2 presents the tasks/decisions covered by a COTS assessment activity framework:

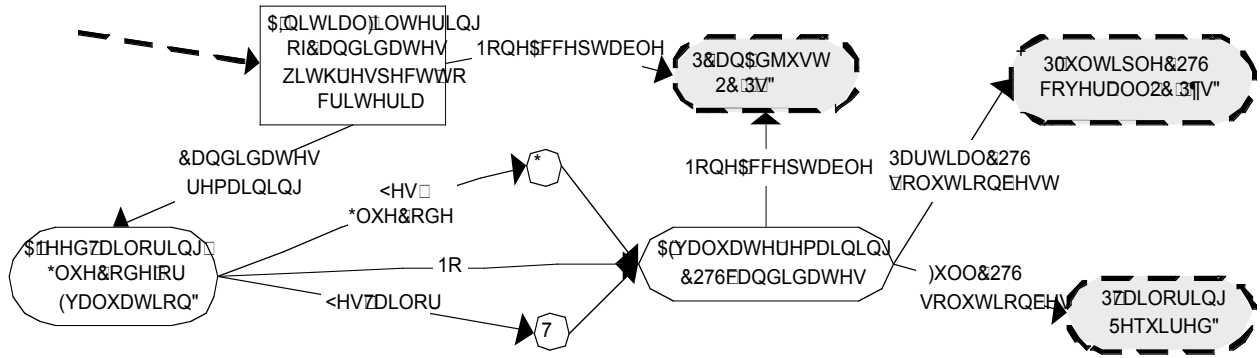


Figure 3-2 COTS Assessment Framework

Entry Conditions for Assessment

The entry condition for assessment assumes that suitable COTS evaluation criterion, their corresponding weights, business scenarios, and COTS candidates are present (starting with the results of Fig. 3.1 decision elements P1 and P2).

Evaluation criteria and weights are established based on stakeholder-negotiated OC&P’s for the system. Stakeholders also agree on the business scenarios to be used for the assessment. The assessment sub-model of COCOTS has collected an extensive list of attributes used in COTS evaluation.

A1: Initial Filtering. Initial assessment tries to quickly filter out the unacceptable COTS packages based on the evaluation criteria. The objective of this activity is to reduce the number of COTS candidates needing to be evaluated in detail. If no available COTS products pass this filtering, this assessment element ends up at the “none acceptable” exit. Some examples of COTS initial filtering techniques are listed below:

- Reference check
- Document review
- Supplier inquiry
- Demo

A2: Tailoring or Glue Code Needed for Evaluation. The remaining COTS candidates from initial filtering will undergo more detailed assessment. To do so, some COTS products need to be tailored (e.g., to assess usability), and some need to be integrated by glue code development (e.g., to assess interoperability).

A3: Detailed Assessment. The focus of detailed assessment is to collect data/information about each COTS candidate against evaluation criteria from pre-designed business scenarios, analyze the data and make decision trade-offs. Some useful techniques are listed here:

1. Use a market watch activity to get the latest COTS information, and collect COTS information from its current users to gain first hand COTS experience from its current user group.
2. Assess vendor supportability to address life cycle issues such as system refresh and maintenance.
3. Develop, instrument, and evaluate prototypes, benchmarks, simulations, or analytic models to analyze key performance parameters and tradeoffs.

A screening matrix or analytic hierarchy process is a useful and common approach to analyze collected evaluation data. The evaluation criteria and COTS candidates work as the columns and rows of the matrix respectively. The final score for a particular COTS candidate is the weighted sum of its points across all of the evaluation criteria. A ranking of all COTS candidates will be produced to help making the COTS decision. However, often a more focused analysis such as a gap analysis or a business case analysis will be needed.

Besides the above major activities taking place during an assessment process element, there are some other management activities that are necessary and even critical to the assessment result. Such management activities are periodic assessment reviews, including the evaluating team, senior management, customers and the key users. The primary tasks for assessment review are to provide feedback to the evaluation process, to negotiate changes of requirements, design and COTS candidates, to adjust and refine the sets of evaluation criteria, weights, and business

scenarios, and make final decisions. The final decisions establish different directions for exiting the COTS assessment process. We have identified the following three exit directions:

1. Full COTS solution is the best, which means there is a single COTS product or a combination of COTS products covering desired OC&P's;
2. A partial COTS solution is the best, which means that COTS product(s) only cover part of the OC&P's, and custom development is needed to meet the gap between COTS and OC&P's;
3. No COTS products are acceptable, which means that pure custom development is the optimal solution, unless the stakeholders are willing to adjust unsatisfied OC&P's.

4.2 Complementary Activity

4.2.1 Market Trend Analyses

In COTS/NDI assessment project, market trend analysis is usually very important in order to properly address market considerations during the assessment. The goals of market trend analysis is to investigate into the market section of your particular domain and find out information such as [4]:

- Standards related to your domain
- Standards based implementations and their attributes. Including business attributes, that can help you solve your problem
- Vendors that are market leaders and their market shares
- Technologies for which products are widely available in the marketplace and the business benefit experienced by the users of a given technology
- Which standards-based implementations work together and an assessment of how well they work.

4.2.2 Product Line Analyses

If the organization that initiates the COTS assessment project is aiming at making appropriate COTS choices that are compatible with the product line architecture, product line analyses is also critical for a successful of COTS assessment activity. The evaluator must pay particular attentions and have insight to the following aspects [5]:

- Qualification of potentially appropriate COTS and NDI components that are fit for use in the system
- Adaptation of components through such means as wrappers, middleware, and other forms of software "glue"
- Assembly of the adapted components to account for the interactions between the adapted components, the architecture, and the middleware
- Updates to the COTS components when new versions are released; decisions about the fit with the current assets and the evolution strategy of the target system; and judgments about the vendor strategy, support, and long-term viability

4.3 Risk Management

Identify and prioritize the top few COTS/NDI assessment risks and plans for resolving them. Update the top-risk list at each progress reporting period. Appendix B provides a list of top risks and corresponding mitigations for COTS assessment projects.

5. Resources (how much)

Summarize the estimated amount of effort, funding, and calendar time that will be required to plan, prepare, execute, analyze, document, review, and refine the assessments. Use and compare at least two bases of estimate (e.g., activity based costing and COCOTS), and identify key assumptions made in performing the estimates.

COTS/NDI Assessment Report (CAR) Document

Preface

The CAR document is reasonably self-contained, but relies on the CAB for detailed background on the project and organizational goals and environment. It also relies on the CAP for details on milestones, budgets, schedules, and risks. Its level of detail is risk-driven, particularly with respect to budgets, schedules and customer needs. Intermediate versions of the CAR should be produced for major milestones in the assessment process.

1. Executive Summary

Summarize the high-level objectives and context of the COTS/NDI assessment, the major results, the conclusions, and the resulting recommendations. Keep the summary on a single page.

2. Purpose, Scope, and Assumptions

The purpose of this document is to

- Summarize the COTS/NDI assessment process;
- Present the major COTS/NDI assessment results and conclusions;
- Make recommendations to the client based on the COTS/NDI assessment results; of the [name of system], and
- Provide substantiation for the results.

The scope of this document covers the COTS/NDI assessment objectives, context, approach, results, conclusions, recommendations, and supporting data, plus [add any other significant topics covered].” Normally, other activities such as a market trend analysis will have separate reports; provide references to those here.

The following assumptions underlie the analysis, results, conclusions, and recommendations:

- [list assumptions]”

3. Assessment Approach

3.1 System Objectives and Context

Briefly summarize the application system objectives, constraints, and priorities. Refer to CAB Sections 2, 3, and 4 for details.

3.2 Assessment Objectives and Approach

Briefly summarize the COTS/NDI assessment objectives and approach. Elaborate somewhat on the Executive Summary, but refer to CAP Section 2.1 and Section 4 for details.

4. Assessment Results

This section can be organized in different ways, depending on the critical assessment issues.

If the assessment involves several relatively independent and compatible types of COTS, primarily organizing by type of COTS works well.

If some assessments involve multiple rounds of downselecting among many COTS/NDI candidates, organizing by round of assessment (level of detail) works well.

If the assessment is dominated by several system-wide issues (performance, scalability, dependability, usability), organizing by issue works well.

Whatever the organization of this section, cover the following elements:

- Types of COTS/NDI products assessed;
- Candidate products assessed for each type;
- Assessment Criteria, Weights, and rating scales;
- Types of assessment used for each criterion (reference check, document review, supplier inquiry, demo, analysis, execution test, prototype application).

- Assessment Scenarios (as appropriate)
- Assessment Ratings and Summaries

4.x Assessment Results-Part X

4.x.1 COTS Assessed

COTS Product	Web Address	Description
eProject	www.eproject.com	
Iplanet package	www.iplanet.com	
blackboard	Learn.usc.edu	

4.x.2 Evaluation Criteria

An example set of evaluation criteria derived from USC COCOTS Assessment component’s attributes list is shown in the following table, the last column presents corresponding weight assigned based on discussion between the client and team members:

No.	Evaluation Criteria – COTS attributes	Weight
1	Inter-component Compatibility	90
2	Product performance	150
3	Functionality:	500
4	Documentation Understandability	60
5	Flexibility:	80
6	Maturity of product	50
7	Vendor support	80
8	Security	150
9	Ease of use	150
10	Training	80
11	Ease of installation/upgrade	60
12	Ease of maintain	60
13	Scalability:	100
14	Vendor viability/stability	60
15	Compatibility with USC IT infrastructure	100
16	Evolution Ability	80
17	Ease of Integration with third-party software	90

You should also describe the quantification method for scoring each criteria and explain briefly why.

The following table shows a template for the COTS product feature checklist. It breaks down the single criteria, “Functionality”, in the above table, into more details in order to better measure the functionality of a COTS product. It can also help doing gap analysis.

Weight	Features	Weight	Second Level Features
0.05	User authentication	0.025	User Login
		0.025	New User Registration
0.1	Create new group	0.1	
0.05	Group Page	0.05	
0.15	chat room	0.03	Easy to use
		0.03	Speed
		0.03	Can the chat content be saved?
		0.03	Support voice chat?
		0.03	Can the chat content be printed out?

0.15	message Board	0.03	Have post/reply/delete function?
		0.03	Only authorized user can use delete function?
		0.02	How many message at most?
		0.05	Support query by name, date, or theme?
		0.02	Can the message/board content be printed out?
0.1	Calendar	0.025	Nice interface?
		0.025	View calendar by month, week, and day?
		0.025	Add/view/delete function?
		0.025	Can the calendar content be printed out?
0.2	Project Management	0.03	Has scheduling function?
		0.03	Has change schedule function?
		0.02	Has task notification function? (new task)
		0.03	Has task progress report function?
		0.03	Has task reminder function? (when login)
		0.03	Has workflow control feature?
		0.03	Has support for printing out task report or what?
0.2	File Management	0.03	Has upload file function?
		0.03	Has download file function?
		0.03	Interface friendly?
		0.03	Had Configuration management support?
		0.02	Has new uploaded file notification function?
		0.02	Only authorized user can use delete function?
		0.02	Maximum file size limitation?
		0.02	Maximum storage space limitation?
	Total	1	

4.X.3 Test procedure (Optional)

Summarize the test procedures and test results performed on COTS based on business scenarios of the target system.

If intensive COTS evaluation testing is performed, prepare an individual testing description document. Please refer to “Appendix C: COTS Evaluation Test Description Document” for example.

4.x.4 Evaluation Results Screen Matrix

eProject							eStudio							eRoom									
Rate-1	Rate-2	Rate-3	Rate-4	Rate-5	Average	Score	Rate-1	Rate-2	Rate-3	Rate-4	Rate-5	Average	Score	Rate-1	Rate-2	Rate-3	Rate-4	Rate-5	Average	Score			
10	10	10	10	10	10	0.25	10	10	10	9	9	9.6	0.24	8	10	9	9	9	9	0.23			
10	10	10	10	10	10	0.25	10	10	10	10	9	9.8	0.25	8	9	9	9	9	8.8	0.22			
9	10	10	10	10	9.8	0.98	9	7	8	8	8	8	0.8	8	8	0	8	8	6.4	0.64			
10	10	10	10	10	10	0.5	8	10	8	9	9	8.8	0.44	6	8	7	8	8	7.4	0.37			
0	0	0	0	0	0	0	6	9	8	8	9	8	0.24	8	10	10	10	8	9.2	0.28			
0	0	0	0	0	0	0	6	7	8	7	7	7	0.21	7	10	9	9	9	8.8	0.26			
0	0	0	0	0	0	0	10	10	10	10	9	9.8	0.29	5	5	5	5	5	5	0.15			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	7	9	10	9	10	9	0.27	4	5	5	5	5	4.8	0.14			
9	9	9	9	10	9.2	0.28	9	9	8	10	9	9	0.27	7	9	8	8	8	8	0.24			
9	8	10	10	9	9.2	0.28	8	7	10	10	8	8.6	0.26	8	9	10	10	8	9	0.27			
10	10	10	10	10	10	0.2	10	9	10	10	7	9.2	0.18	10	9	10	10	9	9.6	0.19			
9	10	10	8	10	9.4	0.47	10	10	9	10	10	9.8	0.49	0	5	0	0	0	1	0.05			
10	10	10	10	10	10	0.2	5	6	5	5	5	5.2	0.1	6	8	5	5	5	5.8	0.12			
8	9	9	8	9	8.6	0.22	10	9	10	9	9	9.4	0.24	8	10	9	10	10	9.4	0.24			
10	10	10	10	10	10	0.25	10	10	10	10	10	10	0.25	10	10	9	9	10	9.6	0.24			
10	10	9	9	10	9.6	0.24	10	10	10	10	10	10	0.25	9	10	9	10	9	9.4	0.24			
10	10	10	10	10	10	0.25	5	6	5	4	5	5	0.13	6	6	6	5	6	5.8	0.15			
9	10	10	10	10	9.8	0.29	10	10	9	10	10	9.8	0.29	8	9	9	9	9	8.8	0.26			
10	10	10	10	10	10	0.3	10	10	9	10	10	9.8	0.29	7	10	8	8	9	8.4	0.25			
9	10	10	10	10	9.8	0.2	9	10	10	9	9	9.4	0.19	9	9	10	10	7	9	0.18			
9	10	10	9	10	9.6	0.29	10	10	9	10	10	9.8	0.29	8	9	9	8	9	8.6	0.26			
10	10	10	9	9	9.6	0.29	3	0	0	0	0	0.6	0.02	0	0	0	0	0	0	0			
8	8	10	8	9	8.6	0.26	7	6	5	7	9	6.8	0.2	9	9	6	7	9	8	0.24			
10	10	10	10	10	10	0.3	7	8	8	8	9	8	0.24	5	7	5	6	5	5.6	0.17			
10	10	10	9	10	9.8	0.29	8	10	9	9	9	9	0.27	10	10	10	9	9	9.6	0.29			
10	10	10	10	10	10	0.3	9	10	9	10	9	9.4	0.28	10	10	10	10	10	10	0.3			
9	9	10	9	10	9.4	0.28	7	9	8	8	9	8.2	0.25	9	9	9	10	9	9.2	0.28			
9	10	10	8	10	9.4	0.28	4	7	0	3	2	3.2	0.1	10	8	10	9	10	9.4	0.28			
10	10	10	10	10	10	0.2	10	9	8	10	9	9.2	0.18	10	8	10	10	8	9.2	0.18			
10	10	10	10	10	10	0.2	10	9	9	10	10	9.6	0.19	9	10	10	10	7	9.2	0.18			
10	10	10	10	10	10	0.2	6	8	8	10	9	8.2	0.16	7	8	8	10	7	8	0.16			
10	10	10	10	10	10	0.2	9	9	9	10	9	9.2	0.18	8	8	7	10	7	8	0.16			
							8.24								8.06								7.21

4.x.5 Business Case Analysis

Establish a business case analysis for each COTS candidate/solution, including the following cost items:

- 4.x.5.1 COTS Ownership Cost
- 4.x.5.2 Development Cost
- 4.x.5.3 Transition Cost
- 4.x.5.4 Operational Cost
- 4.x.5.5 Maintenance Cost

5. Conclusions and Recommendations

In general, it is best to organize these into (conclusion-recommendation) pairs, for example:

- C1. ER Mapper has by far the best performance, but runs only on Windows, failing the acceptable portability criterion. Mr SID is fully portable, and has acceptable performance.
- R1. Use Mr SID for the oversize image viewer function.
- C2. The DBMS assessment is still underway, and Mr SID's interoperability is still uncertain.
- R2. Perform an interoperability assessment between Mr SID and the two DBMS finalists.

Appendix A: COCOTS Assessment Component

1. List of COTS classes:

- 1 Generic Component
- 2 Back office retail
- 3 Communication protocols/packages
- 4 Compilers
- 5 Configuration mgmt/build tools
- 6 Data conversion packages
- 7 Databases
- 8 Device drivers
- 9 Disk arrays
- 10 Emulators
- 11 Engineering tools (req mgmt, design)
- 12 Graphic information system
- 13 GUIs/GUI builders
- 14 Middleware
- 15 Network managers
- 16 Operating systems
- 17 Problem mgmt
- 18 Report generators
- 19 Software process tools
- 20 Telecommunication & infrastructure
- 21 Telemetry Analysis
- 22 Telemetry processing
- 23 Word processing

2. List of COTS attributes and definitions:

CORRECTNESS

Accuracy - The freedom of system output from error.

Correctness - The degree to which a COTS component is free from faults in its specification, design, and implementation.

AVAILABILITY/ROBUSTNESS

Availability - The degree to which a COTS component is operational and accessible when required for use. Often expressed as a probability.

Fail safe - Pertaining to a COTS component that automatically places itself in a safe operating mode in the event of a failure.

Fail soft - Pertaining to a COTS component that continues to provide partial operational capability in the event of certain failures.

Fault tolerance - Pertaining to a COTS component that is able to continue normal operation despite the presence of faults.

Input Error tolerance - The ability of a COTS component to continue normal operation despite the presence of erroneous inputs.

Redundancy - The presence of auxiliary components in a system to perform the same or similar functions as other elements for the purpose of preventing or recovering from failures.

Reliability - The ability of a COTS component to perform its required functions under stated conditions for a specified period of time; the probability that a COTS component will perform its intended functions satisfactorily for a prescribed time and under stipulated conditions.

Robustness - The degree to which a COTS component can function correctly in the presence of invalid inputs or stressful environmental conditions.

Safety - Protection against software or hardware faults that could result in harm to people, data or systems.

SECURITY

Security (Access Related) - the degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.

Security (Sabotage Related) - Protection against exploitable weaknesses that could result in harm to people, data, or systems.

PRODUCT PERFORMANCE

Execution Performance - The degree to which a COTS component performs its functions within given execution timing constraints.

Information/data Capacity - The quantity of information or logical data items that can be stored or maintained by a system or COTS component relative to the expected needs of the users.

Precision - The degree of exactness or discrimination with which a quantity is stated; for example, a precision of 2 decimal places versus a precision of 5 decimal places.

Memory Performance - The degree to which a COTS component performs its functions within given memory constraints (hard storage and/or virtual storage).

Response time - The elapsed time between the end of an inquiry or command to an interactive computer system and the beginning of the system's response.

Throughput - The amount of work that can be performed by a COTS component in a given period of time, for example, number of jobs per day.

UNDERSTANDABILITY

Documentation quality - The degree to which a COTS component contains enough information to explain its objectives, operations, properties and other attributes to be useful in understanding, tailoring, verifying, and operating it.

Simplicity - The degree to which a COTS component has a design and implementation that is straightforward and easy to understand.

Testability - The degree to which a COTS component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

EASE OF USE

Usability/Human Factors - The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.

VERSION COMPATIBILITY

Downward compatibility - Pertaining to software that is compatible with an earlier or less complex version of itself, for example, a COTS component that handles files created by an earlier version of itself.

Upward compatibility - Pertaining to software that is compatible with a later or more complex version of itself, for example, a COTS component that handles files created by a later version of itself.

INTERCOMPONENT COMPATIBILITY

Compatibility with other components - The ability of two or more components to perform their required functions while sharing the same hardware or software environment.

Interoperability - The ability of two or more systems or components to exchange information and to use the information that has been exchanged.

FLEXIBILITY

Extendability - The ease with which features can be added to or around a COTS component in order to increase storage or functional capability.

Flexibility - The ease with which a COTS component can be tailored for use in applications or environments other than those for which it was specifically designed or is normally used.

INSTALLATION/UPGRADE EASE

Installation ease - The ease with which a COTS component can be installed within a hardware or software environment.

Upgrade/refresh ease - The ease with which a new version of a COTS component can be installed within a hardware or software environment.

PORTABILITY

Portability - The ease with which a COTS component can be transferred from one hardware or software environment to another.

FUNCTIONALITY

Functionality - The degree to which a COTS component has the functional capability needed by a user to solve a problem or achieve an objective; a functional capability that must be met or possessed by a COTS component to satisfy a set of requirements.

PRICE

Initial purchase or lease – The upfront cost to buy or lease a COTS component.

Recurring costs – The periodic (usually annual) cost for maintenance and other COTS-related support.

MATURITY

Product Maturity - The length of time that a COTS component has been commercially available and/or the size and diversity of its user base.

Vendor Maturity - The length of time that a vendor has been in the COTS software business and/or the size and diversity of its user base.

VENDOR-SUPPORT

Response time for critical problems - The speed with which critical problems are addressed and solutions are put in place by the vendor.

Support - Responsiveness in answering user questions, and in dealing with user problems in installing, testing, and using the COTS component.

Warranty - The vendor's written guarantee that the product will perform as specified and that instances of non-compliance will be resolved according to a written agreement between the vendor and the buyer.

TRAINING

User training - The degree to which vendor training results in users who are proficient in using the COTS component to solve problems or accomplish objectives.

VENDOR CONCESSIONS

Willingness to escrow source code - Willingness of the vendor to place the source code in the hands of a third-party, thereby providing protection to the procurer in the event that the vendor goes out of business or stops supporting the COTS component.

Willingness to Make Modifications - Willingness to make and maintain procurer-specific modifications to the COTS product, rather than being driven solely by general market demands.

Appendix B: Top Risks and Mitigations for COTS Assessment Projects

Risk	Common Mitigation Plan
1. Requirements changes and mismatches	Prototyping and business case analysis can help to estimate the effect of change and corresponding team effort and schedule needed. Win Win negotiation among all stakeholders must be maintained in each development phase.
2. Many new non technical activities are introduced in the Software Engineering Process	Stakeholders with Domain Expertise, Evaluation expertise must be included in the evaluation process
3. Miss possible COTS candidates within the COTS process	Stay as broad as possible when doing the initial searching for candidates
4. Too much time spent in assessment due to too many requirements and too many COTS candidates	Identify the core ‘showstopper’ requirements and filter all the COTS candidates that do not meet these during the initial assessment and then proceed for a more detailed assessment with the remaining COTS candidates
5. Might not include all key aspects for establishing evaluation criteria set. (Inadequate COTS assessment)	Involve experienced, knowledgeable stakeholders for reviewing evaluation criteria and weight distribution judgments
6. Introducing new COTS candidates is likely and requires re-planning	Develop a contingency plan in cases of addition of a new COTS product. Identify the limits on schedule and budget while making the introduction
7. Faulty Vendor Claims may result in feature loss and/or significant delays	Detailed analysis provides greater assurance of COTS characteristics with respect to vendor documentation (although at significant effort). Detailed assessment beyond literature review or vendor provided documentation should be performed in the form of hands-on experiments and prototyping (especially of core capabilities to be utilized).
8. Ability or willingness of the organization to accept the impact of COTS requirements	The project operational concept must identify such risks and they must be conveyed to the higher management
9. Difficulty in coordinating meetings with key personnel may result in significant delays	The key decision making personnel must be well accounted for the project life cycles. The project manager must make them aware of the approximate time required to be spent by them during the process of assessment etc. The decision making personnel must be kept as minimal as possible
10. Inadequate vendor support may result in significant project delays	The licensing of COTS products must account for vendor support details. In case of contracting labor the developers with experience in using the COTS must be selected
11. COTS package incompatibilities may result in feature loss and significant project delays (Integration Clash)	COTS integration issues must be considered during assessment. The number of COTS products must be kept as minimal as possible.
12. Added complexity of unused COTS features	The number of unused features could be identified and the added complexity because of the presence of such features must be calculated during COTS assessment
13. Overly optimistic expectations of COTS quality attributes	Significant quality features must be tested before selecting COTS products. Special testing packages may be used. Evaluations could be carried out at sites where the COTS is actually being used
14. Overly optimistic COTS package learning curve	An most likely COTS package learning curve must be accounted for during planning the schedule
15. A version upgrade may result in re-tailoring of COTS package	Ensure that the features used to implement the capabilities still exist in the new version before the version upgrade.

Appendix C: COTS Evaluation Test Description Document of xxxx System

1. Test Identification

The products that are going to be tested are two COTS packages, Blackborad (learn.usc.edu) and eProject (www.eproject.com). Both of them are web-based applications that partially satisfy our system requirements, and each has some extra features as well. Our COTS evaluation test has four test procedures which will be performed by four team members and the results of our COTS evaluation tests will work as the basis for our final COTS recommendation.

2. Test Preparation

2.1 Hardware preparation

During cs577b, hand-on experiment was performed to evaluate three COTS candidate: eProject, Blackboard, and iPlanet. For detailed information about these three products, please refer to our COTS dossier. The hand-on experiment is designed and executed to test the functionality and performance of critical business scenarios of each COTS product.

The hardware requirements for each of the three COTS products are shown in the following table:

ID	Package Name	Hardware requirement	Met or not
577b-1	eProject	Computers with the Internet connection for evaluators to get to www.eproject.com website.	Yes. Tests use SAL lab or home PC.
577b-2	Blackboard	Computers with the Internet connection for evaluators to get to learn.usc.edu website.	Yes. Tests use SAL lab or home PC.
577b-3	iPlanet	A server running Sun Soloris OS.	Yes. ISD offered.

2.2 Software preparation

The software requirements for each of the three COTS products are shown in the following table:

ID	Package Name	Software requirement	Met or not
577b-1	eProject	eProject enterprise version 4.0.	Yes. The client helped to negotiate and contract with eProject company to get the three months' test version of it.
577b-2	Blackboard	Blackboard 5.02.	Yes. It is currently running by ISD for course management purpose. No extra cost to get software license for our testing.
577b-3	iPlanet	iPlanet Web Server iPlanet Directory Server iPlanet Collaboration Pack iPlanet Calendar Server	Yes. It has already been bought by ISD.

2.3 Other pre-test preparations

ID	Package Name	Other requirement	Met or not
577b-1	eProject	The testers need authenticated user name and password set-up for the website.	Yes. EProject tech. Representative set up for the test

			group.
577b-2	Blackboard	The testers need authenticated user name and password set-up for the website.	Yes. ISD staff has helped to create a collaboration test group for our team.
577b-3	iPlanet	The four components have to be installed and configured to work together on ISD supported server.	No. Some problem came up during the configuration process. Refer to iPlanet problem report.

3. Test Procedure Specifications

Describe each test case following the template:

Test Case:	1-1
Identifier:	eP-1-1
Test Items:	Create a new project
Test Description :	This test will test the capability of eProject to create a new project. The test will include creating a new project and deleting the project.
Pre-conditions :	The followings will be performed before we can test the item: <ol style="list-style-type: none"> 1. Prepare the information for the new project 2. Click on “My Projects” on the top menu bar 3. Click on the “Create Project” link
Post-conditions	The following functions will be performed: <ol style="list-style-type: none"> 1. Add name of the project 2. Add description of the project 3. Define a group of the project 4. Set the start and end date 5. Set the priority of the project 6. Add the budget 7. Project setting (add modules and set permissions) 8. Check the result after submit the information 9. Delete the project
Input Specifications	The following information will be added to perform the testing: <ol style="list-style-type: none"> 1. Add name of the project <ul style="list-style-type: none"> • Add “My New project” as the name of the project • Also add nothing to test the error message 2. Add description of the project <ul style="list-style-type: none"> • Add any description as the description of the project 3. Define a group of the project <ul style="list-style-type: none"> • In drop-down menu chose “Undefined group” • Also, chose a group in the list, if available 4. Set the start and end date <ul style="list-style-type: none"> • Set a date by click on the pop-up calendar • Set a date by typing a date, “04/02/02” • Type in an illegal date in both fields • Type end date before the start date 5. Set the priority of the project <ul style="list-style-type: none"> • Set it to “High”, “Medium” or “Low” 6. Add the budget <ul style="list-style-type: none"> • Add a number for budget • Add nothing • Add non-digit character 7. Project setting (add modules and set permissions) <ul style="list-style-type: none"> • Check some modules

	<ul style="list-style-type: none"> • Uncheck all modules • Set permission for each module <p>8. Check the result after submit the information</p> <ul style="list-style-type: none"> • Check of the error message and the result are appropriate <p>9. Delete the project</p>
<p>Expected Output Specifications</p>	<p>The following information will be added to perform the testing:</p> <ol style="list-style-type: none"> 1. Add name of the project <ul style="list-style-type: none"> • Add “My New project” as the name of the project <ul style="list-style-type: none"> ○ The result should be ok, the name will be added after submitting the information • Also add nothing to test the error message <ul style="list-style-type: none"> ○ The error message will be shown because the required field is not filled. 2. Add description of the project <ul style="list-style-type: none"> • Add any description as the description of the project <ul style="list-style-type: none"> ○ The information added will be added after submitting. It should be ok if the filed is not filled. 3. Define a group of the project <ul style="list-style-type: none"> • In drop-down menu chose “Undefined group” <ul style="list-style-type: none"> ○ The information will be added. The group will be set in the undefined group. • Also, chose a group in the list, if available <ul style="list-style-type: none"> ○ The information will be added. The group will be 4. Set the start and end date <ul style="list-style-type: none"> • Set a date by click on the pop-up calendar <ul style="list-style-type: none"> ○ The correct-format date will be automatically added to the field after choosing the date from the pop-up calendar • Set a date by typing a date, “04/02/02” <ul style="list-style-type: none"> ○ The information will be added with no error • Type in an illegal format date in both fields <ul style="list-style-type: none"> ○ The error should be detected after submitting the data • Type end date before the start date <ul style="list-style-type: none"> ○ The error message will indicate the error 5. Set the priority of the project <ul style="list-style-type: none"> • Set it to “High”, “Medium” or “Low” <ul style="list-style-type: none"> ○ The priority of the project will set based on the information added 6. Add the budget <ul style="list-style-type: none"> • Add a number for budget <ul style="list-style-type: none"> ○ The information will be added with no error • Add nothing <ul style="list-style-type: none"> ○ Should be ok because it is not required field • Add non-digit character <ul style="list-style-type: none"> ○ The error should be detected 7. Project setting (add modules and set permissions) <ul style="list-style-type: none"> • Check some modules <ul style="list-style-type: none"> ○ Some modules will be added • Uncheck all modules <ul style="list-style-type: none"> ○ No module will be added • Set permission for each module <ul style="list-style-type: none"> ○ The permission properties will be added 8. Check the result after submit the information <ul style="list-style-type: none"> • Check of the error message and the result are appropriate <ul style="list-style-type: none"> ○ The error messages should be appropriate
<p>Pass/Fail Criteria</p>	<p>The following information is the pass/fail criteria for testing:</p> <ol style="list-style-type: none"> 5. Add name of the project <ul style="list-style-type: none"> • The filed should accept any characters for name

	<ul style="list-style-type: none"> • The field is the required field. If the name is not added, the error should be detected. If not, the test fails. <ol style="list-style-type: none"> 6. Add description of the project <ul style="list-style-type: none"> • The field should be able to accept any character for the description • The field is not required to be filled 7. Define a group of the project <ul style="list-style-type: none"> • “Undefined Group” should be the default group • The user can choose the group for the new project from the drop down list. 8. Set the start and end date <ul style="list-style-type: none"> • The pop-up calendar will help the user to choose date. The date will be filled when choose. • The format of the date should be “04/02/02” • An illegal format date has to be detected. • The end date will be before the start date. The error should be detected if the end date is before the start date. 5. Set the priority of the project <ul style="list-style-type: none"> • When set it to “High”, “Medium” or “Low”, the data will be accept. • There should be some default data 6. Add the budget <ul style="list-style-type: none"> • The field will accept only positive numbers • It is not a required field • Non-digit character is not accepted 7. Project setting (add modules and set permissions) <ul style="list-style-type: none"> • When modules added, the project will show modules as added. • There should be some default modules • Set permission for each module will be as set. 8. Check the result after submit the information <ul style="list-style-type: none"> • The error message and the result are appropriate 9. Delete the project <ul style="list-style-type: none"> • The project will be delete after deleting • There should be some prompt message before deleting
Test process	<ol style="list-style-type: none"> 1. User click on “My project” button 2. Click on “Create Project” button 3. Fill out the information needed for creating the project 4. Fill out some illegal information to test the robustness 5. Click “Save” button to create the project 6. Delete the project
Assumptions and constraints	The creator of the project is automatically the project manager. Only project manager will be able to update the project information and delete the project.
Dependencies	The invited members in the Project will be automatically grant the access to the Project.
Traceability	None

4. Test Results

Summarize test results of each test performed following the template:

Test Case:	1
Identifier:	eP-1
Test Items:	Create a new project
Test Result Classification (Pass/Fail):	Pass: The scenario testing result for project creation capability was successful as expected. The test pass both creating a new project manually and using the template.
Problem/Defect Report:	None

Feedback/Comment:	The eProject project creation capability enable user to create a project with ease, both by creating manually and by using a template. The step of creating is very easy A project can be defined however you wish, but each project is a unique entity with its own set of users, documents, tasks, issues, etc. Once you create a new project, you can view it and all of your projects from the My Projects screen. There is no defect or error during the testing. The rating is 9 from 10.
--------------------------	---

5 Test Summary

5.1 Summary

The tests were performed on two COTS packages, Blackborad (learn.usc.edu) and eProject (www.eproject.com). Our COTS evaluation test has four test procedures which will be performed by four team members and the results of our COTS evaluation tests will work as the basis for our final COTS recommendation.

The versions of the two COTS products under testing are eProject enterprise 4.0. and Blackboard 5.0.2.

5.2 Summary of Results and Consequences

Both COTS products have covered some of the functionalities that our system needs, however, the detailed evaluation tests revealed that each of them works well and suitable for only a subset of our targeted system users. Eproject is better suitable for ISD staff, faculty and other professional users since the collaborative project management features it provides is so professional that might not be appropriate for USC common students to fulfill their small scale of team projects. On the other hand, the testing on detailed Blackboard features shows students can collaborate through its chat-room, file sharing, send-email, task, calendar, assignments, etc, to accomplish their team/group-based projects.

5.3 Evaluation

Most of the tests are done successfully, expect that some unavailable and low priority extra features of COTS product. See the test specification.

5.4 Summary of Activities

Total staff that involved in testing are five of team members.

Total machine time is about 120 hours.

Total elapsed time used for each of the major testing activities is about 10 hours.

Appendix D: Effort Report

Team:	Week:	
Name:	ID:	
Management		
Life Cycle Planning		
Changes to plan		(hr)
Additional Contingency Planning		(hr)
Client Interaction		(hr)
Team Interaction		(hr)
Project Web Site		(hr)
ARB Review		(hr)
Inspection and Peer Review		(hr)
Environment and CM		
Environment Preparation		(hr)
Assessment Preparation		
Domain Background Training		(hr)
COTS vendor (tech. people) contact		(hr)
COTS vendor provided training		(hr)
COTS licence negotiation		(hr)
COTS Installation		(hr)
Configuration Management		
COTS documentation maintenance		(hr)
COTS Assessment maintenance		(hr)
COTS Prototype maintenance		(hr)
Requirements		
COTS Initial Assessment(Filtering)		
Initial OC&P's modelling		(hr)
Initial Evaluation Criteria Establishment		(hr)
Maintaining Evaluation Template		(hr)
COTS Initial Assessment Effort		(hr)
Initial Assessment techniques		(give description/explanation about how to perform the initial assessment, not amount of effort)
Number of COTSs initially assessed		(give the number of COTS products initially filtered)
Name of COTSs initially assessed		(give the name of COTS products initially filtered)
COTS Detailed Assessment		
Detailed OC&P's modelling		(hr)
Evaluation Criteria Refinement		(hr)
Detailed Assessment Effort		(hr)
Detailed Assessment Techniques		(give description/explanation about how to perform the detailed assessment, e.g. Prototyping, Testing, Business case analysis, not amount of effort)
Maintaining Evaluation Template		

Guidelines for Developing COTS-Based Applications_v1.0

Number of COTSs detailed assessed		(give the number of COTS products further assessed)
Name of COTSs detailed assessed		(give the name of COTS products initially filtered)
WinWin Negotiations		
COTS Assessment Periodical Report to Client		(hr)
Assessment Re-convergence(e.g. evaluation criteria, OC&P's Revision)		(hr)
Reasons to change, if any		(give description/explanation about why if any changes is introduced to Assessment)
Documenting of OCD		(hr)
Documenting of SSRD		(hr)
Documenting of SSAD		(hr)
Business Case Analysis		(hr)
Market Trend Analysis		(hr)
Design&Implementation		
COTS Tailoring		(hr)
Customizing Parameters		(hr)
Layout GUIs		(hr)
Writing tailoring scripts		(hr)
COTS glue code development		(hr)
Integrating COTS products		(hr)
Integrating COTS products and custom components		(hr)
Integrating Testing		(hr)
Deployment		
Transition and Support Planning		(hr)
User Defined		
User_Def_1		(hr)
Explanation of what User_Def_1 consists in		
User_Def_2		(hr)
Explanation of what User_Def_1 consists in		

Reference

- [1] L. Brownsword, P. Oberndorf, and C. Sledge, "Developing New Processes for COTS-Based Systems," *Software*, July/August 2000, pp. 48-55.
- [2] C. Abts, "*Extending the COCOMO II Software Cost Model to Estimate COTS-Based System Costs PhD Dissertation*," University of Southern California, 2003.
- [3] B. Boehm, D. Port, Y. Yang, J. Bhuta, and C. Abts, "Composable Process Elements for Developing COTS-Based Applications," IEEE/ACM International Symposium on Empirical Software Engineering 2003, Italy, September 2003.
- [4] B. Meyers, and P. Oberndorf, *Managing Software Acquisition: Open Systems and COTS Products*. Addison-Wesley, 2001.
- [5] L. Northrop, *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice*. Version 2.0, <http://www.sei.cmu.edu/plp/companion/COTSutil.htm>
- [6] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison Wesley, 2000.