Using EVMS with COTS-Based Systems

Mary Jo Staley
Patricia Oberndorf
Carol A. Sledge

June 2002

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

TECHNICAL REPORT
CMU/SEI-2002-TR-022
ESC-TR-2002-022

20020724 206
Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2066.

Using EVMS with COTS-Based Systems

CMU/SEI-2002-TR-022
ESC-TR-2002-022

Mary Jo Staley
Patricia Oberndorf
Carol A. Sledge

June 2002

COTS-Based Systems

Unlimited distribution subject to the copyright.
This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Norton L. Compton, Lt Col., USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2002 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is provided, the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).
# Table of Contents

Abstract vii

1 Introduction 1

2 Basics of Earned Value 3

3 COTS-Based Systems and Earned Value in a Complex System Environment 7
   3.1 The Software Life-Cycle Framework 7

4 COTS-Based Systems Activities Mapped to Unified Framework 11

5 The COTS-Based System Work Breakdown Structure 25

6 A Simple COTS-Based System Example Using Product Evaluation as a Stand-Alone Project 31
   6.1 Profile of the Evaluation Project 31
   6.2 Estimating the Effort for CBS Development 32

7 Conclusion 37

Acronym List 39

Appendix: Key Activities by Phase 41

Bibliography 47
List of Figures

Figure 1: Establishing the Baseline 3
Figure 2: Earned Value Elements Plotted 5
Figure 3: Iterative Life-cycle Activities from Royce 8
Figure 4: CBS Activity Sets 11
Figure 5: WBS for WWW Server Evaluation 32
Figure 6: Scheduled Work 33
Figure 7: Earned Value Plan Spreadsheet 34
Figure 8: Time-Phased EV Plan Implemented with Spreadsheet Tool to Collect Measurement 35
Figure 9: Earned Value Plan 35
List of Tables

Table 1: CBS Activities Mapped to Framework 13
Table 2: Software WBS with COTS-Based System Activities 26
Abstract

With the increased use of commercial off-the-shelf (COTS) software products, managers of software development projects must deal with planning and tracking performance of projects that have new challenges and risks. A system developer may be required to integrate multiple COTS products with newly developed custom components and legacy system components. How are these new activities and tasks planned and monitored? Can traditional management methods be used?

Earned Value is a project management tool used extensively to plan and monitor performance against the plan. This paper's focus is on the use of Earned Value in the context of a COTS-Based System (CBS). It's written for an audience already familiar with Earned Value Project Management; only the basic definitions are discussed here with the associated terminology. A bibliography is included, offering good sources for obtaining more in-depth information on Earned Value history and methodology.
1 Introduction

With the increased use of COTS software products, managers of software development projects must deal with planning and tracking performance of projects with new challenges and risks. A system developer may be required to integrate multiple COTS products with newly developed custom components and legacy system components. How are these new activities and tasks planned and monitored? Can traditional management methods such as Earned Value and Work Breakdown Structures be used?

Since Earned Value can be used for any work that can be sized and scheduled, the answer to the question, “Can Earned Value be used for a COTS-based system (CBS)?” is obviously “Yes.” More difficult questions include “What are the CBS-unique activities?” “Where in a development lifecycle do the CBS-unique activities fit?” “How is a time-phased project plan developed with COTS components in the mix of work?”

The SEI CBS initiative has documented the answer to the question, “What are the CBS-unique activities?” in Oberndorf [Oberndorf 00]. In this report we review the basic attributes of Earned Value. Then we map CBS activities to a system lifecycle, selected to illustrate how activities involving COTS products coincide with other defined system development stages and activities. We describe a basis for a lifecycle performance measurement plan and a CBS Work Breakdown Structure (WBS) template, and define a sample Earned Value plan for a typical CBS activity.
2 Basics of Earned Value

Earned Value is a technique that managers can use to control project cost and schedule. The concept of Earned Value is not new. It has been used in manufacturing industry for years, and in 1967 the Department of Defense (DoD) issued the Cost/Schedule Control Systems Criteria (C/SCSC) and mandated their use on systems developed for DoD. In 1997 the DoD approved a set of revised earned value criteria. [DoD 01]. Known as the “Earned Value Management System” (EVMS), it reduces the number of criteria from 35 to 32. With this change, EVMS progressed from being a government requirement to gaining private industry acceptance and use.

To properly use EVMS, a project performance baseline must be established. The elements of a performance baseline are scope, schedule, and cost, as shown in Figure 1.

1. **SCOPE THE WORK**

2. **SCHEDULE THE WORK**

3. **ALLOCATE BUDGETS**

![Diagram of establishing the baseline]

*Figure 1: Establishing the Baseline*
The integration of these elements forms the time-phased baseline with which progress can be tracked and predicted. The development of the project performance baseline is an iterative process for creating a bottom-up detailed plan. The iteration integrates the three elements of an Earned Value plan: scope, schedule and cost.

The scope of the work to be accomplished is usually defined and managed using a Work Breakdown Structure (WBS). A WBS represents all work that is within the scope of the software project. The WBS is represented as a hierarchical chart or an indented list that clearly shows the work, decomposed to three or four levels of detail.

Elements of the WBS are decomposed into manageable pieces called control accounts (CA). Control accounts are composed of work tasks called work packages (WP). A work package has schedule, budget, and organizational responsibility assigned. Frequently, a control account cannot be decomposed into work packages because the account is for future work for which the detail has not evolved. In such cases, Earned Value Planning Packages (PP) are defined instead as decompositions of the control account. A Planning Package evolves with the ongoing addition of detail as information becomes sufficiently available for planning future work. This feature of Earned Value allows for incremental planning, which is essential when developing a COTS project. The use of a CBS Work Breakdown Structure is explored later in this paper.

Once the scope of work is defined and responsibility assigned to an organizational entity, the defined work is planned and scheduled to the performing level; the required resources are estimated and budgets authorized. The sum of all the budgets for all planned work scheduled within a given time period is known as the Budgeted Cost for Work Scheduled (BCWS).

The performance against the plan is determined by calculating the value of the work accomplished at a point in time against the plan to produce the Budgeted Cost of Work Performed (BCWP). The Actual Cost of Work Performed (ACWP) is the summation of the costs actually incurred and recorded in accomplishing all work performed for the time period. The performance against the plan can be calculated with respect to schedule and cost. The difference between the planned value of the work scheduled and the value of the work accomplished for the same time period (BCWP – BCWS) is the schedule variance (SV), which can be used to calculate the percentage of the work a project set out to accomplish that was or was not accomplished in the scheduled time. The cost variance (CV) is defined as the difference between the value of the work accomplished and the actual cost incurred to perform the work (BCWP-ACWP); utilizing this parameter, the percentage of cost overrun or underrun can be calculated. The total estimated work in the plan is the sum of all the budgets and is called the Budget At Completion (BAC), while the Estimate At Completion (EAC) is the projected final cost and is based on a statistical prediction using the performance factors and indices. Figure 2 shows all these Earned Value elements.
Figure 2: Earned Value Elements Plotted
3 COTS-Based Systems and Earned Value in a Complex System Environment

To establish that Earned Value can be employed for a CBS project, it is necessary to think about the activities that are new to the software development process as a result of using COTS products. In addition to just knowing activities, it is helpful to know when in the life of a system's development the activities relevant to the successful selection and implementation of COTS components are performed. Can we take an existing documented life cycle and map these CBS-unique activities into the life cycle with enough definition to create a "seamless" flow with other, more traditionally defined activities of the life cycle?

It is necessary to discuss creating an Earned Value management plan in the context of the life cycle and methodology, since many of the elements of a plan are dependent upon completion of specific artifacts or scheduled milestones of the development or maintenance cycle.

The framework chosen for this work is the iterative life cycle or Unified Framework presented in Walker Royce's *Software Project Management: A Unified Approach* [Royce 98] and *The Unified Software Development Process* by Jacobson, Booch, and Rumbaugh [Jacobson 99]. (See Appendix for a summary.) The Unified Framework allows for incremental planning and promotes planning with a degree of fidelity that matches the knowledge of the project at the time the plan is made. Plans evolve with the evolving system. The following paragraphs define the basics of the life cycle.

3.1 The Software Life-Cycle Framework

The Unified Framework is composed of two stages, four phases, and seven core workflows that are repeated in each iteration within a phase (see Figure 3). The stages are defined as the engineering stage and the production stage. It is during the engineering stage that the builders of the system bring the system to the point of an architectural baseline. Once the decision is made to build the system based upon a selected architecture, the model calls for moving from the engineering to the production stage, at which time the system is constructed and transitioned to the user.

The phases of the life cycle are Inception, Elaboration (together constituting the Engineering Stage), Construction, and Transition (together constituting the Production Stage). A generic
iteration is defined with the core workflows: Management, Requirements, Design, Implementation, Assessment, Deployment, and Environment.

![Diagram of iterative life-cycle activities]

**Figure 3: Iterative Life-cycle Activities from Royce**  
[Royce 99]

The development of a system progresses through all the phases. Within a phase the work is performed by iterating through the workflows—one iteration can in some ways be compared to the traditional waterfall concept, although perhaps a better description is to call an iteration a mini-project. The focus of work related to the workflows varies depending upon the phase of development of the system. However, for each phase, work is defined for each workflow. For example, during the inception phase emphasis is on requirements, while in the construction phase emphasis is on implementation, but both workflows are part of both phases.

The major work product for a core workflow is a defined model of the system, where each model is simply the work products that represents some aspect of the system (e.g., design, code). All of the models for all of the core workflows are fleshed out as the development of the system progresses through the phases. The following paragraphs define at a high level the characteristics of the models.
These models are described in the Appendix. The data in the tables in the Appendix are summarized from both books used as references for the Unified Framework [Jacobson 99, Royce 98]. The mapping of CBS activities to Unified stages and activities contained in Section 4 makes more sense if we understand what is happening in the various phases with the custom development components of our complex system that also has COTS components. The Appendix provides a high-level chart for each of the phases, showing the activities within each phase, along with the key deliverables. The engineering workflows (i.e., those workflows that are not management activities) and artifacts are fully covered in The Unified Software Development Process [Jacobson 99].
4 COTS-Based Systems Activities Mapped to Unified Framework

A set of activities that are new or changed for COTS-based systems has been compiled [Oberndorf 00]. These activities are categorized in “activity areas”; within each area are clusters of activities referred to as “activity sets,” as depicted in Figure 4.

![Diagram showing activity areas for COTS-Based Systems](image)

**Figure 4:** CBS Activity Sets
The activity areas are Engineering, Business, Contract, and Program-Wide. For this report, we have mapped the activities in the activity sets to the appropriate phases of the unified framework. This mapping is shown in Table 1.

A CBS activity may span multiple phases of the life cycle. This is accounted for in Table 1 where certain "activity cells" cross multiple "phase columns" of the table.
<table>
<thead>
<tr>
<th>CBS Area</th>
<th>Activity Set</th>
<th>Engineering Stage</th>
<th>Production Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Vendor Relationships</td>
<td>Explore formal and informal vendor relationships.</td>
<td>Establish and maintain vendor relationships.</td>
</tr>
<tr>
<td>Business</td>
<td>Vendor Relationships ARTIFACTS</td>
<td>License agreements</td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>COTS Business Case</td>
<td>Develop Preliminary business case.</td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>COTS Business Case</td>
<td>Obtain and document stakeholder concurrence with critical success factors.</td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>COTS Business Case</td>
<td>Make the COTS Business Case recommendation.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: CBS Activities Mapped to Framework
<table>
<thead>
<tr>
<th>CBS Area</th>
<th>Activity Set</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>COTS Business Case ARTIFACTS</td>
<td>• Preliminary COTS Business Case</td>
<td>• Completed COTS Business Case</td>
<td>• Revised COTS Business Case</td>
<td>• Documented sensitivity analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Documented critical success factors.</td>
<td>• Written critical success factors.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>Inter-government¹ Supplier Relationships from Acquirers' Point of Reference</td>
<td>• Explore formal and informal relationships with inter-government suppliers and their critical vendors.</td>
<td>Establish and maintain formal and informal relationships with inter-government suppliers and their critical vendors.</td>
<td>• Establish and maintain formal and informal relationships with inter-government suppliers and their critical vendors.</td>
<td>• Periodically re-assess relationships with suppliers and their vendors.</td>
</tr>
</tbody>
</table>

¹  Inter-government supplier relationships could be replaced with inter-organizational supplier relationships in a commercial setting.
<table>
<thead>
<tr>
<th>COTS Based Systems Activities</th>
<th>Engineering Stage</th>
<th>Production Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CBS Area</strong></td>
<td><strong>Activity Set</strong></td>
<td><strong>Inception</strong></td>
</tr>
<tr>
<td>Business</td>
<td>Inter-government Supplier Relationships from Acquirers' Point of Reference ARTIFACTS</td>
<td>• License agreements with vendors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Document describing relationship with inter-government suppliers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Strategy document for inter-government relationships</td>
</tr>
<tr>
<td>Business</td>
<td>Inter-government Supplier Relationships from Supplier Point of View ARTIFACTS</td>
<td>• Document a strategy for NDI.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Establish user support groups.</td>
</tr>
<tr>
<td>Business</td>
<td>Inter-government Supplier Relationships from Supplier Point of View ARTIFACTS</td>
<td>• Strategy Document for marketing, maintaining and supporting NDI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COTS Based Systems Activities</td>
<td>Engineering Stage</td>
<td>Production Stage</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>CBS Area</td>
<td>Activity Set</td>
<td>Inception</td>
</tr>
</tbody>
</table>
| Business | COTS Cost Estimation | - Establish models and techniques for COTS cost estimate.  
- Estimate preliminary COTS costs. Collect actual cost data. | - Refine COTS cost models.  
- Refine COTS costs estimate.  
- Collect Actual cost data. | | |
| Business | COTS Costs Estimation ARTIFACTS | - Cost estimation models and techniques  
- Preliminary cost estimates  
- Actual costs collected | - COTS Cost estimation models and techniques  
- Cost estimates  
- Actual costs collected | | |
<p>| Contract | License Negotiation | | Investigate and negotiate Licenses. | | |
| Contract | License Negotiation ARTIFACTS | | | Licenses Agreements | |</p>
<table>
<thead>
<tr>
<th>Contract</th>
<th>Contract Requirements ARTIFACTS</th>
<th>Documented contract requirements</th>
<th>Documented impact analysis of contract changes with respect to COTS approach.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract</td>
<td>Solicitation</td>
<td></td>
<td>Prepare acquirer estimates and criteria for products and services that include incentives.</td>
</tr>
<tr>
<td></td>
<td>Solicitation ARTIFACTS</td>
<td></td>
<td>COTS relative parts of the RFP</td>
</tr>
</tbody>
</table>
| Engineering   | System Context [denotes the collection of requirements (functional and non-functional, including the context of their end-user processes) and other constraints such as cost and schedule] | • Identify constraints.  
• Identify key user processes and product mismatches.  
• Participate in process/product trade-off negotiations. | • Identify constraints.  
• Identify key user processes and product mismatches.  
• Promulgate tradeoffs. |
<table>
<thead>
<tr>
<th>Engineering</th>
<th>System Context ARTIFACTS</th>
<th>System context tradeoff decisions document including rationale and commitments</th>
<th>System context tradeoff decisions document including rationale and commitments</th>
<th>Revised affected documents (e.g. ORD, test plans)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering</td>
<td>Architecture and Design ARTIFACTS</td>
<td>Preliminary CBS architecture and design Alternative architecture and design</td>
<td>Architectural prototype Revised CBS architecture and design</td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>Marketplace ARTIFACTS</td>
<td>Engineering</td>
<td>Construction</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Information dis-</td>
<td></td>
<td>Prototype to discover product and system characteristics.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>semination plan</td>
<td></td>
<td>Perform preliminary integration of selected components.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Marketplace in-</td>
<td></td>
<td>• Tailor and integrate selected components.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>formation (e.g., re-</td>
<td></td>
<td>• Develop glue code.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ports, database)</td>
<td></td>
<td>• Test components and system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Revised information</td>
<td></td>
<td>• Receive and analyze product upgrades; incorporate as appropriate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dissemination plan</td>
<td></td>
<td>• Receive and analyze product upgrades; incorporate as appropriate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Revised Marketplace</td>
<td></td>
<td>• Isolate, analyze and resolve system faults.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>information (e.g.,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reports, database)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>Construction ARTIFACTS</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>Prototypes</td>
<td>CBS Detailed Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prototype results reports</td>
<td>Prototype and accompanying documentation (e.g. user manuals, install procedures)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developed Source Code Test Plans Accompanying documents Initial system implementation Refined CBS architecture and detailed design Produce upgrade impact analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Revised: Source Code Test Plans Accompanying documents Initial system implementation Refined CBS architecture and detailed design Produce upgrade impact analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>Configuration Management</td>
<td>Identify configuration baselines.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Revise configuration baseline. Release new system versions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release new system versions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>Configuration Management ARTIFACTS</td>
<td>Configuration baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated configuration baselines System Release</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Release</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Engineering | Deployment and Sustainment | • Determine and implement strategies for system deployment and end user support of COTS products.  
  • Manage incorporation of new product release and new products including management of licenses. | • Revise and implement strategies for system deployment and end user support of COTS products.  
  • Manage incorporation of new product release and new products including management of licenses. |
| Engineering | Deployment and Sustainment ARTIFACTS | • Documented strategies and plans  
  • License management plan  
  • Supplier release and implementation plan  
  • “Refreshed” prototype | • Revised strategies and plans  
  • Revised License management plan  
  • Revised Supplier release and implementation plan  
  • “Refreshed” prototype |
| Engineering | Evaluation | Plan and execute evaluation tasks. |
| Engineering | Evaluation ARTIFACTS | • Evaluation Plan  
  • Evaluation Results |
<table>
<thead>
<tr>
<th>Program Wide</th>
<th>Cultural Transition ARTIFACTS</th>
<th>Program Wide</th>
<th>Information Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cultural Transition</td>
<td></td>
<td>Collect and share CBS information.</td>
</tr>
<tr>
<td></td>
<td>• Assess readiness (including skill set required) to transition to CBS and provide training.</td>
<td>• Revise readiness assessment and training plan.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Develop and implement strategy for accomplishing the transition.</td>
<td>• Implement and revise strategy for accomplishing the transition.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Initial CBS needs analysis</td>
<td>• Revised CBS needs analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• COTS Training Plan</td>
<td>• Revised COTS Training Plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CBS transition strategy and plan</td>
<td>• Revised CBS transition strategy and plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Transition lessons learned</td>
<td>• Expanded Transition lessons learned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Incentives and awards for use of CBS</td>
<td>• Incentives and awards for use of CBS</td>
<td></td>
</tr>
</tbody>
</table>
| Program Wide | Information Sharing ARTIFACTS | • Strategy and plan for CBS information sharing including models of usage, dissemination and maintenance  
• Mechanisms for collecting and sharing CBS information  
• Collected and shared information | • Revised strategy and plan for CBS information sharing including models of usage, dissemination and maintenance  
• Maintained mechanisms for collecting and sharing CBS information  
• Collected and shared information |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Wide</td>
<td>CBS Tradeoffs ARTIFACTS</td>
<td>Make CBS Tradeoffs.</td>
<td>Tradeoff analyses including rationale and decisions</td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Tradeoffs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Tradeoffs ARTIFACTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Strategy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Risk Management ARTIFACTS</td>
<td>Identify and manage CBS risks.</td>
<td>CBS risks identification and analysis</td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Risk Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Wide</td>
<td>CBS Risk Management ARTIFACTS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5 The COTS-Based System Work Breakdown Structure

According to Royce, traditional WBSs are prematurely structured around the system design. This makes for a plan that is difficult and expensive to change. A WBS that is organized around the process rather than the design of the system can evolve without breaking the entire structure. An evolutionary WBS, then, is one that accounts for the evolution of the WBS itself and contains planning tasks in each major phase. These planning tasks provide the structure to elaborate the next phase without breaking the WBS. The first-level elements of an evolutionary WBS are the core workflows, the second-level elements are defined for each of the phases within the life cycle, and the third-level elements are defined for the activities within the phase that produce the artifacts.

The default WBS with the Unified Framework [Royce 98] description is used as the starting point in the following CBS WBS (Table 2). The CBS activities have been inserted into the template and italicized. Each system is unique, and the WBS is specific to the system. This generic example provides a framework that can be exploited and tailored for a real project.

Observe that planning occurs throughout the life cycle; each phase creates a plan for the next phase, which is then detailed when the phase begins. This type of management cycle can use the concepts of Earned Value to build the first plan with Planning Packages that are elaborated when the information is available to make a more detailed plan.
Table 2: Software WBS with COTS-Based System Activities

1. Management

1.1. Inception phase management
1.1.1. Business case development
1.1.2. COTS business case development
   1.1.2.1. Preliminary COTS business case development
   1.1.2.2. Critical COTS success factors development
1.1.3. Vendor relationships
   1.1.3.1. Vendor relationships exploration
   1.1.3.2. License agreements exploration
1.1.4. Inter-government supplier relationships
   1.1.4.1. Inter-government supplier relationship exploration
   1.1.4.2. Inter-government agreements exploration
1.1.5. COTS cost estimates
   1.1.5.1. COTS cost estimate cost model and technique establishment
1.1.6. COTS license negotiation
1.1.7. Contract requirements establishment
1.1.8. Elaboration phase release specification
   1.1.8.1. Plan CBS strategy (including vendor relationship strategy)
1.1.9. Elaboration phase WBS baselining
1.1.10. Software development plan
1.1.11. Inception phase project control and status assessments

1.2. Elaboration phase management

1.2.1. Construction phase release specification
1.2.2. COTS business case development
   1.2.2.1. COTS business case recommendation
   1.2.2.2. Critical COTS success factors agreement
1.2.3. Vendor relationships establishment and maintenance
   1.2.3.1. License agreements and documented relationships maintenance
1.2.4. Inter-government supplier relationships maintenance
1.2.5. COTS cost estimate
   1.2.5.1. COTS cost estimate refinement
   1.2.5.2. COTS cost data collection
1.2.6. CBS Contract Solicitation
   1.2.6.1. Preparation of estimates for products and services
1.2.7. Construction phase WBS baselining
1.2.8. Elaboration phase project control and status assessments

1.3. Construction phase management

1.3.1. Transition phase planning
1.3.2. COTS business case development
   1.3.2.1. Sensitivity analysis factors monitoring
   1.3.2.2. COTS business case revision
1.3.3. Vendor relationships maintenance
   1.3.3.1. Formal and informal relationships strategy re-assessment
   1.3.3.2. License agreement maintenance
1.3.4. Inter-government supplier relationships maintenance
   1.3.4.1. Formal and informal relationships strategy re-assessment
1.3.5. CBS culture transition
   1.3.5.1. Readiness assessment
   1.3.5.2. Strategy for transition development and implementation
1.3.6. Transition phase WBS baselining
1.3.7. Construction phase project control and status assessments

1.4. Transition phase management
1.4.1. Next generation planning
   1.4.1.1. CBS migration strategy and plan
   1.4.1.2. Migration lessons learned

1.4.2. Transition phase project controls and status assessments
1.4.3. COTS business case maintenance and revision
1.4.4. Intergovernment supplier relationships maintenance
1.4.5. Vendor relationships maintenance
1.4.6. CBS culture transition for CBS development
   1.4.6.1. Initial CBS needs analysis
   1.4.6.2. COTS training plan
   1.4.6.3. CBS test plans development

2. Environment
2.1. Inception phase environment specification
   2.1.1. CBS development/integration environment specification
   2.1.2. CBS test-bed specification
   2.1.3. CBS information sharing repository and data specification

2.2. Elaboration phase environment baselining
   2.2.1. Development environment installation and administration
      2.2.1.1. CBS development/integration environment installation and administration
      2.2.1.2. CBS test-bed installation and administration
      2.2.1.3. CBS information repository installation, data population, and administration

   2.2.2. Development environment integration and custom toolsmithing
      2.2.2.1. CBS development/integration environment integration, custom toolsmithing, and maintenance
      2.2.2.2. CBS test-bed environment integration, custom toolsmithing, and maintenance
      2.2.2.3. CBS information sharing repository integration, custom toolsmithing, and maintenance

   2.2.3. Software change order database formulation
      2.2.3.1. CBS development/integration change order database formulation

   2.2.4. Configuration management installation and administration
      2.2.4.1. CBS development/integration environment configuration management installation and administration

2.3. Construction phase environment maintenance
2.3.1. Development environment installation and administration
   2.3.1.1. CBS development/integration environment installation, administration, and maintenance
   2.3.1.2. CBS test-bed administration and maintenance
   2.3.1.3. CBS information sharing repository installation, administration, and maintenance

2.3.2. Software change order databases maintenance
   2.3.2.1. CBS development/integration installation maintenance

2.3.3. Configuration management installation and administration
   2.3.3.1. CBS development/integration installation and maintenance

2.4. Transition phase environment maintenance
2.4.1. Development environment maintenance and administration
2.4.1.1. CBS development/integration installation maintenance
2.4.1.2. CBS test-bed maintenance
2.4.1.3. CBS information repository maintenance
2.4.2. Software change order database maintenance
  2.4.2.1. CBS development/integration installation maintenance
2.4.3. Maintenance environment packaging and transition
  2.4.3.1. CBS development/integration installation packaging and transition
  2.4.3.2. CBS test-bed packaging and transition
  2.4.3.3. CBS information repository packaging and transition
2.4.4. Configuration management installation and administration
  2.4.4.1. CBS development/integration installation maintenance

3. Requirements
3.1. Inception phase requirements development
  3.1.1. Vision specification
  3.1.2. CBS system context\textsuperscript{2} tradeoff
  3.1.3. CBS Marketplace
    3.1.3.1. Marketplace information creation, dissemination, refresh
  3.1.4. Use case modeling
    3.1.4.1. CBS feasibility demonstration prototype
3.2. Elaboration phase requirements baselining
  3.2.1. Vision baselining
  3.2.2. CBS process and product tradeoff negotiation
  3.2.3. CBS process and product tradeoff baselining
  3.2.4. Use case model baselining
3.3. Construction phase requirements maintenance
3.4. Transition phase requirements maintenance

4. Design
4.1. Inception phase architecture prototyping
  4.1.1. CBS architecture and design
    4.1.1.1. Preliminary CBS architecture development
    4.1.1.2. Alternative CBS architecture development
  4.1.2. CBS Marketplace
    4.1.2.1. Marketplace information creation, dissemination, refresh
4.2. Elaboration phase architecture baselining
  4.2.1. Architecture design modeling
    4.2.1.1. CBS architectural prototyping
    4.2.1.2. System tradeoff decision and commitments
    4.2.1.3. COTS product selections
  4.2.2. Software architecture description
    4.2.2.1. CBS architecture and design description
4.3. Construction phase design modeling
  4.3.1. Architecture design model maintenance
  4.3.2. CBS architecture and design
    4.3.2.1. CBS architecture and design maintenance
    4.3.2.2. COTS product upgrade impact analysis
  4.3.3. Component design modeling
4.4. Transition phase design maintenance

\textsuperscript{2} System context denotes the collection of requirements (functional and nonfunctional, including the context of their end-user processes) and other constraints such as cost and schedule.
5. Implementation
5.1. Inception phase component prototyping
   5.1.1. CBS construction
      5.1.1.1. COTS prototypes development
5.2. Elaboration phase component implementation
   5.2.1. Critical component coding demonstration integration
      5.2.1.1. COTS demonstration prototype integration
      5.2.1.2. COTS product selection
5.3. Construction phase component implementation
   5.3.1. Initial release(s) component coding and stand-alone testing
      5.3.1.1. COTS product tailoring
      5.3.1.2. COTS products upgrades receipt and analysis
      5.3.1.3. CBS component glue code implementation
   5.3.2. Alpha release component coding and stand-alone testing
      5.3.2.1. COTS product tailoring
      5.3.2.2. COTS product upgrades receipt and analysis
      5.3.2.3. CBS component glue code implementation
   5.3.3. Beta release component coding and stand-alone testing
      5.3.3.1. COTS product tailoring
      5.3.3.2. COTS product upgrades receipt and analysis
      5.3.3.3. CBS component glue code implementation
   5.3.4. Component maintenance
      5.3.4.1. COTS product upgrades analysis
      5.3.4.2. COTS product upgrades configuration management
5.4. Transition phase component maintenance
   5.4.1. CBS deployment and sustainment
      5.4.1.1. New COTS product release management
      5.4.1.2. System deployment and end user support strategies implementation

6. Assessment
6.1. Inception phase assessment planning
6.2. Elaboration phase assessment
   6.2.1. Test modeling
   6.2.2. Architecture test scenario implementation
   6.2.3. Demonstration assessment and release descriptions
6.3. Construction phase assessment
   6.3.1. Initial release assessment and release description
   6.3.2. Alpha release assessment and release description
   6.3.3. Beta release assessment and release description
6.4. Transition phase assessment
   6.4.1. System release assessment and release descriptions

7. Deployment
7.1. Inception phase deployment planning
   7.1.1. CBS deployment and sustainment
      7.1.1.1. CBS deployment/support strategy development
      7.1.1.2. Supplier release and implementation plan development
      7.1.1.3. License management plan development
7.2. Elaboration phase deployment planning
   7.2.1. CBS deployment and sustainment
      7.2.1.1. CBS deployment strategies revision and implementation
7.2.1.2. Supplier release and implementation plan revision
7.2.1.3. License management plan revision

7.3. Construction phase deployment
7.3.1. User manual baselining
7.3.2. CBS deployment and sustainment
7.3.2.1. New product release and license management
7.3.2.2. CBS implementation strategies implementation

7.4. Transition phase deployment
7.4.1. Product transition to user
7.4.2. CBS deployment and sustainment
7.4.2.1. CBS end-user support management
6 A Simple COTS-Based System Example Using Product Evaluation as a Stand-Alone Project

Our premise is that if work can be decomposed into manageable pieces that can be estimated for effort and schedule, then Earned Value techniques can be applied. The following example shows an Earned Value plan for a small product evaluation effort in a CBS project. Desktop tools that included spreadsheets and Microsoft Project were used for this example.

The process used a survey of the marketplace for a web browser suitable for a large Management and Control System. The goal of the evaluation process was to compare features of candidate products against the set of requirements. The selection process included a task to map the features of each candidate product against the requirements. This work package is called "Develop Feature Maps" in the WBS. A team at the SEI performed this COTS evaluation work. At the time the project was executed, it was not managed using Earned Value. But we have been able to reconstruct the work packages of the project from historical data to build this plan. The example is not a case study: the WBS items represent the actual work breakdown that was used; the other parameters have been added for illustration purposes.

6.1 Profile of the Evaluation Project

An evaluation process was followed that consisted of work and work products defined by the Work Breakdown Structure in Figure 5. The WBS is illustrated here as an indented list. It should be noted that this exercise was not embedded in a project that was planned against the Unified Framework. As can be seen in the activities chart in Table 1, an evaluation activity for CBS can occur in any phase of a project. These WBS elements for the evaluation (shown here at Level 3) would then be at Level 4 or Level 5 of a larger project WBS.

1. WWW Server Evaluation
   1.1. Conduct Market Survey
       1.1.1. Identify Candidates
       1.1.2. Develop Product Categories
       1.1.3. Develop Feature Maps
   1.2. Develop Evaluation Criteria
       1.2.1. Develop Evaluation Criteria Check-lists
   1.3. Develop Assessment Plan
       1.3.1. Plan Assessment Technique
6.2 Estimating the Effort for CBS Development

A complete Earned Value plan using the WWW Server Evaluation task elements is shown in as an example. Once resources were assigned to the schedule, the tasks were scheduled using Microsoft Project. The task resource usage report from Microsoft Project supplies the time-phased data (effort in hours), which can then be input into an Excel workbook, which is used to track the progress.

The Earned Value plan is captured in a spreadsheet (Figure 7) that has entries for the plan (BCWS), the earned value (BCWP), and the actual cost (ACWP). Progress against the plan can be entered into the sheet on a weekly basis (Figure 8) and plotted as shown in Figure 9. Many tools are available to plan and track a project using EVMS; the job can also be done with a suite of desktop office products.
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WWW Server Evaluation</td>
<td>70d</td>
</tr>
<tr>
<td>2</td>
<td>Conduct Market Survey</td>
<td>12d</td>
</tr>
<tr>
<td>3</td>
<td>Identify Candidates</td>
<td>5d</td>
</tr>
<tr>
<td>4</td>
<td>Develop Product Categories</td>
<td>30d</td>
</tr>
<tr>
<td>5</td>
<td>Develop Feature Maps</td>
<td>40d</td>
</tr>
<tr>
<td>6</td>
<td>Develop Evaluation Criteria</td>
<td>50d</td>
</tr>
<tr>
<td>7</td>
<td>Compile Checklist</td>
<td>50d</td>
</tr>
<tr>
<td>8</td>
<td>Develop Assessment Plan</td>
<td>13d</td>
</tr>
<tr>
<td>9</td>
<td>Plan Assessment Technique</td>
<td>60d</td>
</tr>
<tr>
<td>10</td>
<td>Develop Assessment Schedule</td>
<td>70d</td>
</tr>
<tr>
<td>11</td>
<td>Conduct Assessment</td>
<td>27d</td>
</tr>
<tr>
<td>12</td>
<td>Vendor Literature</td>
<td>8d</td>
</tr>
<tr>
<td>13</td>
<td>Model Problem</td>
<td>90d</td>
</tr>
<tr>
<td>14</td>
<td>Product Profile</td>
<td>100d</td>
</tr>
<tr>
<td>15</td>
<td>Synthesize Results</td>
<td>130d</td>
</tr>
</tbody>
</table>

**Figure 6: Scheduled Work**
<table>
<thead>
<tr>
<th>Work Components</th>
<th>Est Sloc</th>
<th>Other Hours</th>
<th>Earnable Units</th>
<th>&lt;Milestone 1&gt;</th>
<th>EV</th>
<th>&lt;Milestone 2&gt;</th>
<th>EV</th>
<th>&lt;Milestone 3&gt;</th>
<th>EV</th>
<th>Total Earned</th>
<th>% Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earnable percentages by Milestones</td>
<td>30%</td>
<td>40%</td>
<td>30%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component #1 Conduct Market Survey</td>
<td>0</td>
<td>96</td>
<td>96</td>
<td>100%</td>
<td>28.8</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>28.8</td>
<td>30.00%</td>
</tr>
<tr>
<td>Identify Candidates</td>
<td>40</td>
<td>40</td>
<td>1</td>
<td>12.0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>12.0</td>
<td>30.00%</td>
</tr>
<tr>
<td>Develop Product Categories</td>
<td>24</td>
<td>24</td>
<td>1</td>
<td>7.2</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>7.2</td>
<td>30.00%</td>
</tr>
<tr>
<td>Develop Feature Maps</td>
<td>32</td>
<td>32</td>
<td>1</td>
<td>9.6</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>9.6</td>
<td>30.00%</td>
</tr>
<tr>
<td>Component #2 Develop Evaluation Criteria</td>
<td>0</td>
<td>40</td>
<td>40</td>
<td>50%</td>
<td>6.0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>6.0</td>
<td>15.00%</td>
</tr>
<tr>
<td>Develop Evaluation Criteria Check Lists</td>
<td>40</td>
<td>40</td>
<td>0.5</td>
<td>6.0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>6.0</td>
<td>15.00%</td>
</tr>
<tr>
<td>Component #3 Develop Assessment Plan</td>
<td>0</td>
<td>104</td>
<td>104</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Plan Assessment Technique</td>
<td>48</td>
<td>48</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Develop Assessment Schedule</td>
<td>56</td>
<td>56</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Component #4 Conduct Assessment</td>
<td>0</td>
<td>216</td>
<td>216</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Vendor Literature</td>
<td>64</td>
<td>64</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Model Problem</td>
<td>72</td>
<td>72</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Product Profile</td>
<td>80</td>
<td>80</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Component #5 Synthesize Results</td>
<td>0</td>
<td>192</td>
<td>192</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Compile Assessment Results</td>
<td>176</td>
<td>176</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Conduct Review</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>648</td>
<td>648</td>
<td>18%</td>
<td>34.8</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>34.8</td>
<td>5.37%</td>
</tr>
</tbody>
</table>

Figure 7: Earned Value Plan Spreadsheet
<table>
<thead>
<tr>
<th>Week Beginning</th>
<th>4/4</th>
<th>4/11</th>
<th>4/18</th>
<th>4/25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week End</td>
<td>4/10</td>
<td>4/17</td>
<td>4/24</td>
<td>5/1</td>
</tr>
<tr>
<td>Task BCWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component #1</td>
<td>0</td>
<td>32</td>
<td>72</td>
<td>96</td>
</tr>
<tr>
<td>Conduct Market Survey</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task ACWP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Task BCWP</td>
<td>0</td>
<td>19.68</td>
<td>28.8</td>
<td>0</td>
</tr>
<tr>
<td>Task BCWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component #2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>Develop Evaluation Criteria</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task ACWP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Task BCWP</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Task BCWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component #3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Develop Assessment Plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task ACWP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Task BCWP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 8:** Time-Phased EV Plan Implemented with Spreadsheet Tool to Collect Measurement

**Figure 9:** Earned Value Plan
7 Conclusion

We opened this paper by answering the question, "Can Earned Value be used in a COTS-based system development?" with a "yes." The work reported in this paper confirms that "yes" is the correct answer. We have illustrated in the preceding sections both why the answer is "yes" and how this application of EVM can be accomplished.

There are many COTS-unique activities that must be integrated into our traditional approaches if we are to succeed with COTS-based systems. Earned Value has features, such as the planning packages, that will make it easier to plan an evolving development lifecycle. In addition, the traditional Work Breakdown Structure can be composed of elements that reflect the process rather than the system design. A process-based WBS is particularly advantageous for COTS-based systems: the change of a COTS product or technology can affect the design of the system, which would disturb the whole management process if the WBS were design based.
Acronym List

ACWP, actual cost of work performed
BAC, budget at completion
BCWP, budgeted cost of work performed
BCWS, budgeted cost for work scheduled
CA, control account
CBS, COTS-based system
COTS, commercial off-the-shelf
CV, cost variance
EAC, estimate at completion
EV, earned value
EVMS, earned value management system
IOC, initial operational capability
MOU, memorandum of understanding
NDI, non-developmental item
ORD, operational requirements document
PP, (earned value) planning package
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFP</td>
<td>request for proposal</td>
</tr>
<tr>
<td>SV</td>
<td>scheduled variance</td>
</tr>
<tr>
<td>WBS</td>
<td>work breakdown structure</td>
</tr>
<tr>
<td>WP</td>
<td>work package</td>
</tr>
</tbody>
</table>
Appendix: Key Activities by Phase

Adapted from *The Unified Software Development Process* by Jacobson, Booch, and Rumbaugh [Jacobson 99]

<table>
<thead>
<tr>
<th>Inception Phase Activities</th>
<th>Key Deliverables of Inception Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Description and goals:</strong> The goals of the inception phase is to establish the business case, scope the system, sketch an architecture, identify critical risks, develop a proof of concept prototype.</td>
<td></td>
</tr>
</tbody>
</table>
| • **Requirements:** Identify 50% of use-cases analyze/prototype small percentage for proof of concept during this phase.  
  1. List candidate requirements for system feature list.  
  2. Understand system context.  
  3. Capture pertinent functional requirements.  
  4. Capture related nonfunctional requirements. |
| • **Analysis:** Build initial analysis model that will be the basis of the early design model. This model reveals the shared resources among the selected use-cases for this phase.  
  1. Analyze a use-case.  
  2. Build simple analysis model showing shared resources among selected use-cases. |
| • **Design:** Sketch a design model for the candidate architecture.  
  1. Develop initial design model.  
  2. Identify interfaces between subsystems/classes. |
| • **Implementation:** In most cases the implementation flow is not required/used in the inception phase. |
| • **Test:** Small amount of activity in the test flow for the inception phase if any. |
| • First draft of the business case  
• First version of a business model – sets context of the system  
• A feature list  
• First draft of a candidate architecture description  
• Proof of concept exploratory prototype – demonstrating use of the system  
• Initial risk list and use-case ranking  
• Beginning of plan for entire system  
• First cut of use-case model, analysis model, and design model |
Develop some tentative test plans.
- Select the development environment.
- Develop the initial business case.
### Elaboration Phase

#### Elaboration Phase Activities

**General Description and goals:** Iterate to capture about 80% or the requirements and detail a small percentage to develop the executable architecture baseline.

- **Requirements:** Goal is to capture about 80% of the requirements, implement and test scenarios to develop the architectural baseline.
  1. Find use-cases necessary for architectural baseline – detail only these cases.
  2. Prototype user-interfaces.
  3. Prioritize use-cases.
  4. Detail use cases needed to understand requirements.
- **Analysis:** Build on the analysis model that was begun in the Inception Phase. Perform the analysis activities for use cases that are architecturally significant. These activities include analyze a use case, analyze a class, and analyze a package.
- **Design:** Design the architecturally significant use cases, classes, and subsystems. This usually means less than 10% of the use-case mass.
- **Implementation:** Implement and test the components that were designed. The results are the architectural baseline.
- **Test:** Plan and execute tests that ascertain that the components on all levels work for the architectural baseline. Components are tested as they become available and integrated for the build test.
- **Make the business case.**
- **Plan the construction phase.**

#### Key Deliverables of Elaboration Phase

- Complete business model
- New version of all models
- Executable architectural baseline
- Architecture description
- Updated risk list
- Project plan for construction and transition phases
- Preliminary user manual (optional)
- Completed business case
<table>
<thead>
<tr>
<th>Construction Phase Activities</th>
<th>Key Deliverables for Construction Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Description and goals:</strong> Take the architectural baseline through iterations to develop a software product ready for the initial operational baseline.</td>
<td>• The executable software</td>
</tr>
<tr>
<td>• <strong>Requirements:</strong> Perform the requirements capture for the small % (hopefully &lt;20%) that was not identified and detailed in the elaboration phase.</td>
<td>• All models/artifacts of the system</td>
</tr>
<tr>
<td>• <strong>Analysis:</strong> Perform analysis steps to extend the model to include the addition requirements identified in this phase.</td>
<td>• Maintained architecture description</td>
</tr>
<tr>
<td>• <strong>Design:</strong> Design and implement the remaining 90% of the use cases – those that were not implemented to develop the architectural baseline.</td>
<td>• Preliminary user manual</td>
</tr>
<tr>
<td>• <strong>Test:</strong> Design and perform test cases to test the system. This is performed on a build basis.</td>
<td>• Business Case – reflecting situation at the end of the phase</td>
</tr>
<tr>
<td>• Prepare first cut of user manuals.</td>
<td></td>
</tr>
<tr>
<td>• Plan the transition phase.</td>
<td></td>
</tr>
</tbody>
</table>
## Transition Phase

<table>
<thead>
<tr>
<th>Transition Phase Activities</th>
<th>Key Deliverables for Construction Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Description and goals: Establish the product in the operational environment. The activity is low in all the core workflows for the transition phase. The core flows are concerned with responding to feedback to correct problems. Other parallel activities are performed as listed below.</td>
<td>- The executable software, installation software</td>
</tr>
<tr>
<td>• Prepare the actual Beta Release from the IOC of the construction phase.</td>
<td>• Legal documents, license documents, waivers</td>
</tr>
<tr>
<td>• Install at site.</td>
<td>• All models completed for the baseline</td>
</tr>
<tr>
<td>• Act on feedback to correct defects.</td>
<td>• Complete architecture description</td>
</tr>
<tr>
<td>• Complete the artifacts/manuals.</td>
<td>• End-user, operator, system administrator manuals</td>
</tr>
<tr>
<td>• Determine when project ends.</td>
<td>• Customer support references</td>
</tr>
<tr>
<td>• Assess the transition phase.</td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY
   (Leave Blank)

2. REPORT DATE
   June 2002

3. REPORT TYPE AND DATES COVERED
   Final

4. TITLE AND SUBTITLE
   Using EVMS with COTS-Based Systems

5. FUNDING NUMBERS
   F19628-00-C-0003

6. AUTHOR(S)
   Mary Jo Staley, Patricia Oberdorf, Carol A. Sledge

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
   Software Engineering Institute
   Carnegie Mellon University
   Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
   CMU/SEI-2002-TR-022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
   HQ ESC/XPK
   5 Eglin Street
   Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
    ESC-TR-2002-022

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT
    Unclassified/Unlimited, DTIC, NTIS

12B. DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)
    With the increased use of commercial off-the-shelf (COTS) software products, managers of software development projects must plan and track performance of projects that have new challenges and risks. A system developer may be required to integrate multiple COTS products with newly developed custom components and legacy system components. How are these new activities and tasks planned and monitored? Can traditional management methods be used?

    Earned Value is a project management tool used extensively to plan and monitor performance against the plan. This paper’s focus is on the use of Earned Value in the context of a COTS-Based System (CBS). It’s written for an audience already familiar with Earned Value Project Management; only the basic definitions are discussed here with the associated terminology. A bibliography is included, offering good sources for obtaining more in-depth information on Earned Value history and methodology.

14. SUBJECT TERMS
    Earned Value Management System, EVMS, COTS, Commercial Off-The-Shelf Software, COTS-Based System, CBS, Unified Framework

15. NUMBER OF PAGES
    57

16. PRICE CODE

17. SECURITY CLASSIFICATION OF
    REPORT
    Unclassified

18. SECURITY CLASSIFICATION OF
    THIS PAGE
    Unclassified

19. SECURITY CLASSIFICATION OF
    ABSTRACT
    Unclassified

20. LIMITATION OF ABSTRACT
    UL