

COCOTS Software Integration Cost Model: An Overview

Barry W. Boehm, Ph.D.

Center for Software Engineering
University of Southern California
941 W. 37th Place, SAL 328
Los Angeles, CA 90089-0781
+1-213-740-5703
boehm@sunset.usc.edu

Christopher M. Abts, M.S.

Center for Software Engineering
University of Southern California
941 W. 37th Place, SAL 328
Los Angeles, CA 90089-0781
+1-213-740-6470
cabts@sunset.usc.edu

Elizabeth K. Bailey, Ph.D.

Software Metrics, Inc.
4345 High Ridge Road
Haymarket, Virginia 20169
+ 1-703-754-0115
bkbailey@erols.com

Acknowledgement: appreciation is extended to Mr. A. Winsor Brown, USC-CSE, for arranging the opportunity to participate in CSS '98.

Presentation Abstract

Keywords

COTS, Cost Models, Software Integration

INTRODUCTION

Software engineering in a fully connected world relies increasingly on the integration and tailoring of commercial-off-the-shelf (COTS) software components. This pre-existing software is from commercial vendors who supply self-contained off-the-shelf components that can be plugged into a larger software system to provide capability that would otherwise have to be custom built. The two primary distinguishing characteristics of this COTS software are 1) that its source code is not available to the application developer, and 2) that its evolution is not under the control of the application developer.

The rationale for using COTS based components is that they will involve less development time by taking advantage of existing, market proven, vendor supported products, thereby reducing overall system costs. But because of the two defining characteristics noted above (lack of access to product source code, and lack of control over product evolution), there is a trade-off in using the COTS approach in that new software development time can indeed be reduced, but generally at the cost of an *increase* in software component integration work. COTS software also brings with it a host of unique risks quite different from those associated with software developed in-house.

Included among those factors which should be examined when determining the true cost of integrating a COTS software component into a larger system are not only the traditional costs associated with new software development such as the cost of requirements definition, design, code, test, and software maintenance, but also now the costs of licensing and redistribution rights, royalties, effort needed to understand the COTS software, pre-integration assessment and evaluation, post-integration certification of compliance with mission critical or safety critical requirements, indemnification against faults or damage caused by vendor supplied components, and costs incurred due to incompatibilities with other needed software and/or hardware.

In light of these factors, the first step in developing the COCOTS model was to develop the basic model form and an accompanying prototype tool with the ultimate goal of being able to reasonably and consistently predict the cost of a given COTS software integration effort.

The most significant factors driving COTS integration costs have been identified and mathematical forms for a set of four submodels incorporating those costs have been generated. In its current form COCOTS offers insight into the development costs of using COTS components. With extensions planned in the near future intended to address the entire system lifecycle, including acquisition and O&M costs, COCOTS is on its way to capturing *all* significant costs associated with using COTS software.