COCOTS:
Constructive COTS Integration Cost Model

Center for Software Engineering
University of Southern California

Points of Contact at USC-CSE

Christopher Abts (primary graduate researcher)……(213) 740-6470
Ms. Jennifer Browning (CSE Office Administrator)…..…….. (213) 740-5703
Dr. Barry W. Boehm (CSE Director)…….(213) 740-8163
USC Center for Software Engineering FAX line……..……...(213) 740-4927
COTS Software Research Effort E-Mail…..……*cots-info@sunset.usc.edu*
World Wide Web page……..……*http://sunset.usc.edu/cots-research*

COTS Software Research Effort
Overview Statement

Introduction

The **Office of Naval Research** is providing funding to the **Center for Software Engineering** at the **University of Southern California** (USC-CSE) to further develop a model for the estimation of the costs associated with the use of **commercial-off-the-shelf (COTS) software** in the creation of new software systems. This effort has been on-going since 1995, and has had the past sponsorship of other organizations, including the **Air Force Electronic Systems Center**, and the **Federal Aviation Administration** (sponsorship still on-going), and with the cooperation of the **Software Engineering Institute** at Carnegie Mellon University.

The **Constructive COTS Integration Cost Model (COCOTS)** effort is under the leadership of Dr. Barry W. Boehm, Director of the Center for Software Engineering at USC, and is related to an effort currently underway at USC to develop and refine **COCOMO II**, a modern version of the well-known **Constructive Cost Model (COCOMO)**[1] software cost estimation model.

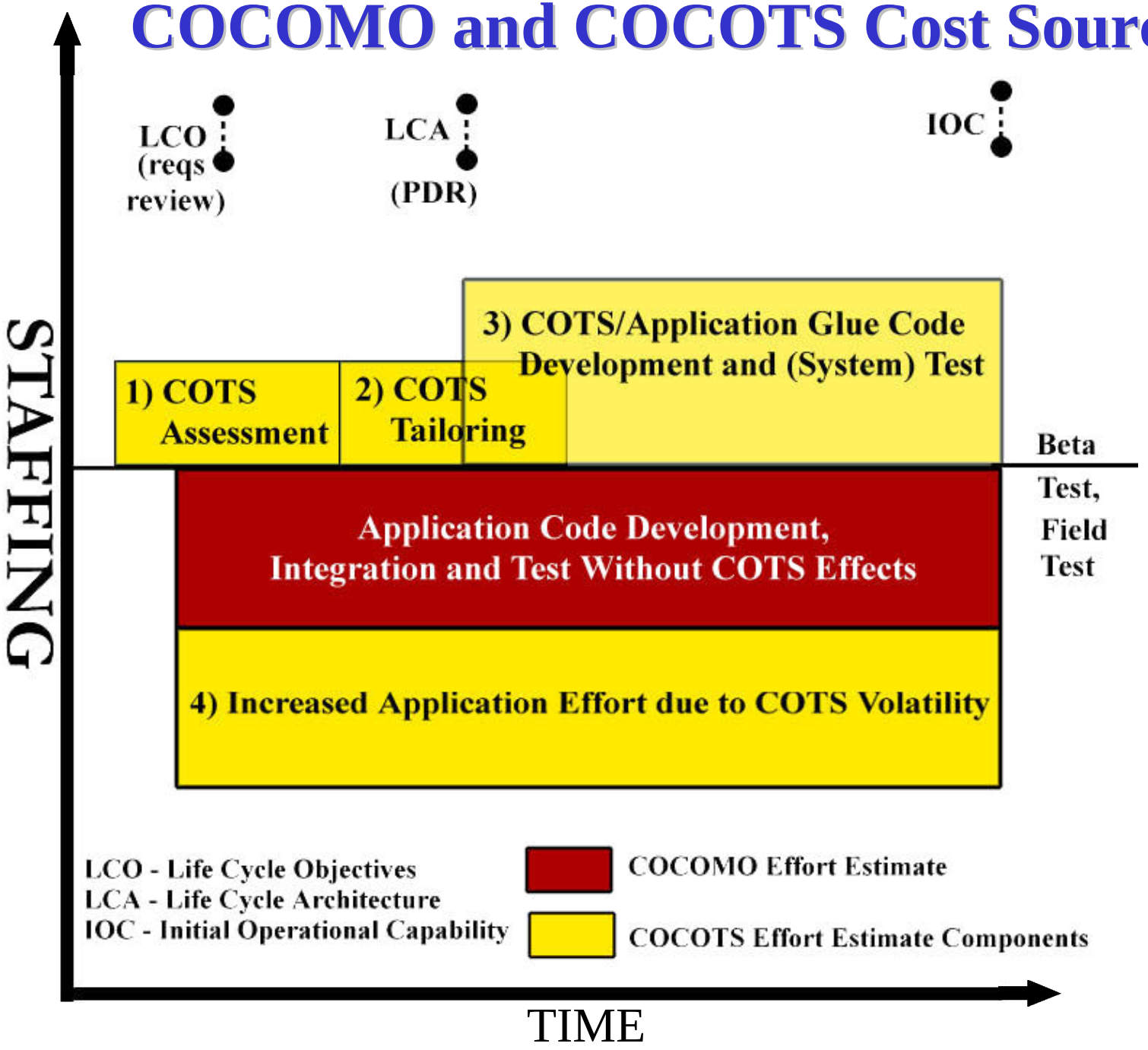
One of the more significant changes in software development practice over the past twenty years is the greatly increased emphasis being placed on building systems incorporating pre-existing software in order to keep overall development costs as low as possible. One source of pre-existing software is commercial vendors who supply self-contained off-the-shelf components which can be plugged into a larger software system to provide capability which would otherwise have to be custom built. The primary distinguishing characteristics of COTS software are that its source code is not available to the application developer, and that its evolution is not under the control of the application developer.

The COCOTS effort to date has identified four primary sources of COTS integration effort: 1) COTS assessment; 2) COTS tailoring; 3) glue code design, development, test, and integration; and 4) added application development effort due to COTS volatility. The effort has also identified candidate models to estimate these sources of effort. The parameters of these models, and the resulting project effort amounts, are described in a COCOTS Data Collection Form and Guidelines document.

Cooperation from the software development community is being solicited to provide data via these forms and guidelines. Such data is needed to test hypotheses and to guide and validate the cost model's formulation. In return for cooperation in gathering data, the cost model will be open and made available to the public, and workshops on its use and refinement will be conducted with the data providers.

[1] COnstructive COst MOdel (COCOMO) is defined in Software Engineering Economics by Barry W. Boehm, Prentice Hall, 1981.

COCOMO and COCOTS Cost Sources



The USC-CSE COTS Integration Cost Estimation
Glue Code Model Equations:

$$PM = A \times \left[Size' \right]^{B} \times \prod_{i=1}^{14} EM_i$$

where

$$A = 12.0$$

$$B = 1.00 + (0.04 \times SF)$$

$$Size' = KSLOC \times \left(1 + \frac{BRAK}{100} \right)$$

Symbol	Description
A	Constant, provisionally set to 12.0
BRAK	Breakage: Percentage of COTS glue code thrown away due to requirements volatility
KSLOC	Size of COTS component glue code expressed in thousands of source lines of code
PM	Person Months of estimated integration effort
EM	Effort Multipliers: ACIEP, ACIPC, AXICP, APCON, ACPMT, ACSEW, APCPX, ACPPS, ACPTD, APVOL, ACREL, AACPX, ACPER, ASPRT
SF	Scale Factor: AAREN

Table 2 - Definition of symbols appearing in the COTS integration cost estimation model.

Glue Code Model Drivers

Personnel Drivers

1) ACIEP - COTS/NDI Integrator Experience with Product
2) ACIPC - COTS/NDI Integrator Personnel Capability
3) AXCIP - Integrator Experience with COTS/NDI Integration Processes
4) APCON - Integrator Personnel Continuity

COTS Component Drivers

5) ACPMT - COTS/NDI Product Maturity
6) ACSEW - COTS/NDI Supplier Product Extension Willingness
7) APCPX - COTS/NDI Product Interface Complexity
8) ACPPS - COTS/NDI Supplier Product Support
9) ACPTD - COTS/NDI Supplier Provided Training and Documentation
10) APVOL - COTS/NDI Product Volatility

Application/System Drivers

11) ACREL - Constraints on Application System/Subsystem Reliability
12) AACPX - Application Interface Complexity
13) ACPER - Constraints on COTS/NDI Technical Performance
14) ASPRT - Application System Portability

Nonlinear Scale Factor

1) AAREN - Application Architectural Engineering