
 University of Southern California  
Center for Software Engineering

---

# **Overview: USC-SEI-CeBASE Workshop on COTS-Based Systems**

**Barry Boehm, USC-CSE  
February 7, 2001**

2/7/01 ©USC-CSE 1


 University of Southern California  
Center for Software Engineering

---

## **Outline**

- **USC-CSE Highlights, 2000**
- **USC-CSE Affiliates and Calendar**
- **Objectives of This Workshop**
- **Candidate Working Groups and Procedures**
- **Wednesday Agenda**
  - **Hello from Jack Ferguson**

2/7/01 ©USC-CSE 2




University of Southern California  
Center for Software Engineering

---

## USC-CSE Highlights, 2000

- **New Affiliates:** Auto Club of So. Calif., Galorath, Group Systems.com, IBM, JPL, Marotz
- **New Ph.D.'s:** Alex Egyed (Teknowledge), Jongmoon Baik (Motorola)
- **Boehm honorary Sc.D. (UMass), INCOSE Fellow, IEEE Mills Award**
- **COCOMO II book and CD**
- **Commercially-based EasyWinWin (GroupSystems.com)**
- **NSF-ITR CeBASE grant with UMaryland, UNebraska, Mississippi State U.**

2/7/01
©USC-CSE
3




University of Southern California  
Center for Software Engineering

---

## USC-CSE Affiliates (33)

- **Commercial Industry (16)**
  - Automobile Club of Southern California, C-Bridge, EDS, Fidelity Group, Galorath, Group Systems.Com, Hughes, IBM, Lucent, Marotz, Microsoft, Motorola, Rational, Sun, Telcordia, Xerox
- **Aerospace Industry (9)**
  - Boeing, Draper Labs, GDE Systems, Litton, Lockheed Martin, Northrop Grumman, Raytheon, SAIC, TRW
- **Government (3)**
  - FAA, US Army Research Labs, US Army TACOM
- **FFRDC's and Consortia (4)**
  - Aerospace, JPL, SEI, SPC
- **International (1)**
  - Chung-Ang U. (Korea)

2/7/01
©USC-CSE
4




University of Southern California  
Center for Software Engineering

## USC-CSE Affiliates' Calendar

June 22, 2000	Easy WinWin Web Seminar
July 25-26, 2000	Easy WinWin Hands-on Tutorial
July 27, 2000	Tutorial: Transitioning to the CMMI via MBASE
August 24-25, 2000	Software Engineering Internship Workshop
September 13-15, 2000	Workshop: Spiral Development in the DoD (Washington DC; with SEI)
October 24-27, 2000	COCOMO/Software Cost Modeling Forum and Workshop
February 6-9, 2001	Annual Research Review, COTS-Based Systems Workshop (with SEI, CeBASE)
February 21-23, 2001	Ground Systems Architecture Workshop (with Aerospace, SEI)
February 21, 2001	LA SPIN, Ron Kohl, COTS-Based Systems Processes
March 28, 2001	LA SPIN, High Dependability Computing
May 2001	Annual Affiliates' Renewal
May-June 2001	Rapid Value/RUP/MBASE Seminar (with C-Bridge, Rational)

2/7/01
©USC-CSE
5



University of Southern California  
Center for Software Engineering

## Center for Empirically-Based Software Engineering (CeBASE) Strategic Vision

**Strategic Framework**

Strategic Process:	Experience Factory
Tailoring G/L:	Goal-Model-Question-Metric
Tactical Process:	Model Integration (MBASE); WinWin Spiral

**Empirical Methods**

<b>Quantitative</b> Experimental Observational Analysis Parametric Models Dynamic Models Pareto 80-20 Relationships	<b>Qualitative</b> Ethnographic Surveys, Assessments Critical Success Factors Root Cause Analysis
--	---

**Experience Base (Context; Results)**

Project, Context Attributes  
 Empirical Results; References  
 Implications and Recommended Practices  
 Experience Feedback Comments

• Initial foci: COTS-based systems; Defect reduction

2/7/01
©USC-CSE
6

**USC**  
**C S E** University of Southern California  
Center for Software Engineering

## Top 10 Risk Items: 1989 and 1995

1989	1995
1. Personnel shortfalls	1. Personnel shortfalls
2. Schedules and budgets	2. Schedules, budgets, process
3. Wrong software functions	3. COTS, external components
4. Wrong user interface	4. Requirements mismatch
5. Gold plating	5. User interface mismatch
6. Requirements changes	6. Architecture, performance, quality
7. Externally-furnished components	7. Requirements changes
8. Externally-performed tasks	8. Legacy software
9. Real-time performance	9. Externally-performed tasks
10. Straining computer science	10. Straining computer science

2/7/01
©USC-CSE
7


**USC**  
**C S E** University of Southern California  
Center for Software Engineering

## CeBASE Software Defect Reduction Top-10 List

- <http://www.cebase.org>

1. Finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase.
2. About 40-50% of the effort on current software projects is spent on avoidable rework.
3. About 80% of the avoidable rework comes from 20% of the defects.
4. About 80% of the defects come from 20% of the modules and about half the modules are defect free.
5. About 90% of the downtime comes from at most 10% of the defects.
6. Peer reviews catch 60% of the defects.
7. Perspective-based reviews catch 35% more defects than non-directed reviews.
8. Disciplined personal practices can reduce defect introduction rates by up to 75%.
9. All other things being equal, it costs 50% more per source instruction to develop high-dependability software products than to develop low-dependability software products. However, the investment is more than worth it if significant operations and maintenance costs are involved.
10. About 40-50% of user programs have nontrivial defects.

2/7/01
©USC-CSE
8




University of Southern California  
Center for Software Engineering

## Prospective CeBASE CBS Top-10 List

1. The main sources of CBS-specific development effort are Glue Code, COTS Tailoring, COTS Assessment and COTS Volatility (COCOTS database)
2. CBS projects can easily overrun budgets and schedules by factors of 4 (Garlan et al., 1995; USC term papers)
3. CBS effort distributions vary by CBS type (COCOTS database)
4. Hypothesis: Most glue code effort levels can be predicted reasonably well by parametric models (COCOTS data)
5. Hypothesis: CBS costs can be estimated via architectural mismatch analysis (UMD, USC)
6. Hypothesis: CBS evolution costs dominate CBS development costs (Lockheed Martin)
7. Hypothesis: Glue-code inspection yields will be lower than new-code inspection yields (UMD)
8. Hypothesis: CBS costs vary as the square of the number of independently-developed COTS products (USC, UMD)
9. Hypothesis: CBS projects are twice as likely to be totaled as non-CBS projects (UMD, USC)
10. Hypothesis: Half of the totaled CBS projects result from trusting in vendor vaporware (UMD, USC).

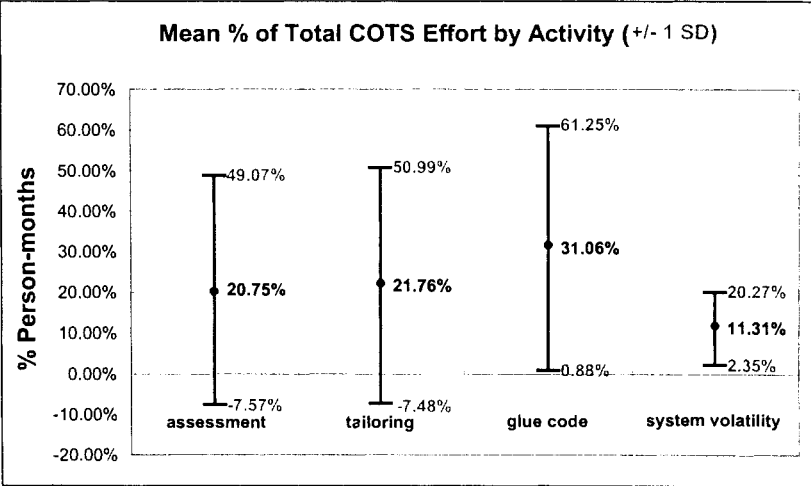
2/7/01
©USC-CSE
9



University of Southern California  
Center for Software Engineering

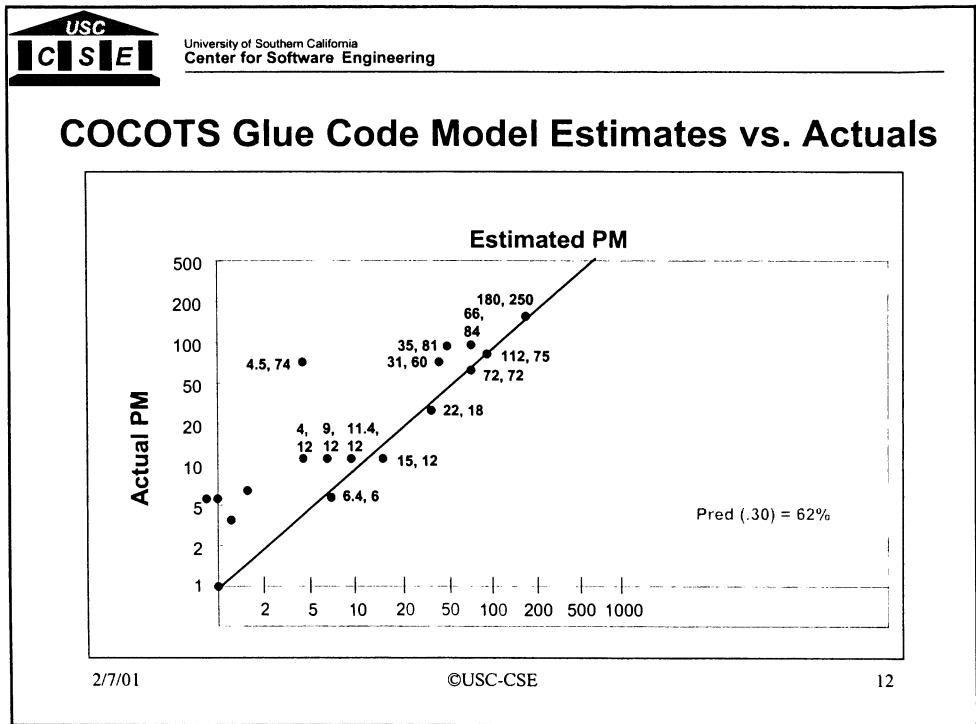
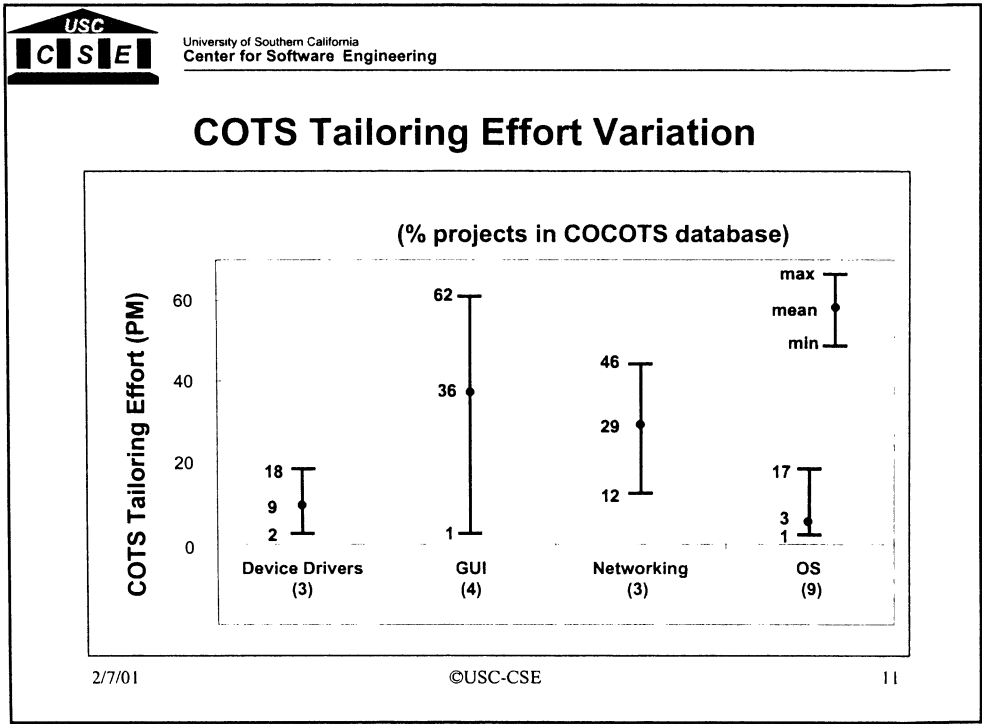
## Data Highlights


**Mean % of Total COTS Effort by Activity (+/- 1 SD)**



Activity	Mean %	+1 SD	-1 SD
assessment	20.75%	49.07%	-7.57%
tailoring	21.76%	50.99%	-7.48%
glue code	31.06%	61.25%	-0.88%
system volatility	11.31%	20.27%	-2.35%

2/7/01
©USC-CSE
10






University of Southern California  
Center for Software Engineering

---

## Workshop Objectives

- **Address key COTS-based system challenges and opportunities**
- **Assess new directions in COTS-based system development and evolution**
- **Provide guidelines for USC-CSE/SEI/CeBASE research, Affiliate activities**
  - Needs, priorities, risks, opportunities
- **Stimulate further USC-CSE/SEI/CeBASE/Affiliate collaboration**

2/7/01
©USC-CSE
13




University of Southern California  
Center for Software Engineering

---

## Top Level Agenda

- **Wed: 8 am - 4 pm: CBS Presentations**
  - 4 - 5 pm: Working groups context setting
  - 5 - 7 pm: Reception
- **Thurs: Working groups**
  - 12:00-2:00 pm: Lunch and plenary session
- **Fri. am: Working group reports; USC-CSE response; wrap-up**
  - End by 11:30 am

2/7/01
©USC-CSE
14




University of Southern California  
Center for Software Engineering

---

## Proposed Working Groups (Coordinators; Scribe)

1. CBS Process Issues (Carney, Port; Al-Said)
2. CBS Architecture Issues (Seacord, Medvidovic; Roshandel)
3. CBS Post-Deployment Issues (Oberndorf, Reifer, Betsy Clark)
4. CBS Metrics and Predictive Models (Abts, Brad Clark)
5. CBS Dependability Issues (Basili, Brown; Lee)

2/7/01 ©USC-CSE 15



University of Southern California  
Center for Software Engineering

---

## Working Group Guidelines

- **Product:** briefing, preferably with notes and priorities
- **Topics should include:**
  - Most critical issues in area
  - Most promising opportunities
  - Results from prioritization surveys
  - Research suggestions: general, CSE/SEI/CeBASE, CSE Affiliates

2/7/01 ©USC-CSE 16