CeBASE and Ground System Architectures

Barry Boehm, USC/CeBASE*
February 21, 2001

*Center for Empirically-Based Software Engineering
http://www.cebase.org
Outline

• CeBASE overview
• Initial Results
  – Relevance to Ground System Architectures
• Experience Factory Overview
  – Methods
  – Results at UMD/SEI, USC/TRW
• Proposed Application to DoD Software Intensive Systems
CeBASE Context

• Research center sponsored by
  – NSF Information Technology Program
• Co-Directors:
  – Victor Basili (UMaryland), Barry Boehm (USC)
• Co-PI’s:
  – Marvin Zelkowitz (UMaryland), Rayford Vaughn (MSU), Forrest Shull (FC-MD), Dan Port (USC), Ann Majchrzak (USC), Scott Henninger (UNL)
• Initial 2-year funding: $2.4 M
Center for Empirically-Based Software Engineering (CeBASE) Strategic Vision

**Strategic Framework**
- **Strategic Process:** Experience Factory
- **Tailoring Guidelines:** Goal-Model-Question-Metric
- **Tactical Process:** Model Integration (MBASE); WinWin Spiral

**Empirical Methods**

<table>
<thead>
<tr>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>Ethnographic</td>
</tr>
<tr>
<td>Observational Analysis</td>
<td>Surveys, Assessments</td>
</tr>
<tr>
<td>Parametric Models</td>
<td>Critical Success Factors</td>
</tr>
<tr>
<td>Dynamic Models</td>
<td>Root Cause Analysis</td>
</tr>
<tr>
<td>Pareto 80-20 Relationships</td>
<td></td>
</tr>
</tbody>
</table>

**Experience Base (Context; Results)**
- Project, Context Attributes
- Empirical Results; References
- Implications and Recommended Practices
- Experience Feedback Comments

- Initial foci: COTS-based systems; Defect reduction
CeBASE Results Chain

Construct knowledge-based (KB) framework

Integrate UMD, USC data, models

Develop CeBASE portal

Initial KB form

Compatible data, models

Access facilitation

Populate KB in COTS-Based Systems (CBS) Defect Reduction (DR) areas

Empirical KB content

Initial KB, empirical techniques

Wide, easy access to CeBASE KB:
Empirical best practices, models, data
Empirical research tools and techniques

Growing KB of empirical best practices models in CBS, DR areas

New results

Growing KB, empirical techniques

Growing community of CeBASE users, empirical researchers; KB content

Continued funding

Expand to other collaborators, areas, best practices

Expanded results

CeBASE usage cost-benefit justification

Outreach and technology transition activities:
Contact, awareness, understanding, trial use, limited adoption, institutionalization

Much more predictable development of both rapid and robust software

Critical-mass community of empirical SW researchers; scientific understanding of software phenomenology

Growing KB of empirical best practices models in CBS, DR areas

Growing community of CeBASE users, empirical researchers; KB content
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.
COCOTS Database Highlights

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

- Assessment: 49.07% (±7.57%)
- Tailoring: 50.99% (±7.48%)
- Glue Code: 61.25% (±0.88%)
- System Volatility: 20.27% (±11.31%)
- 0.00% (±2.35%)

% Person-months
COTS Tailoring Effort Variation

(# projects in COCOTS database)

COTS Tailoring Effort (PM)

Device Drivers (3)
GUI (4)
Networking (3)
OS (9)

max
mean
min
COCOTS Glue Code Model Estimates vs. Actuals

Estimated PM

Actual PM

Pred (.30) = 62%
Outline

• CeBASE overview
• Initial Results
  – Relevance to Ground System Architectures
• Experience Factory Overview
  – Methods
  – Results at UMD/SEI, USC/TRW
• Proposed Application to DoD Software Intensive Systems
The Experience Factory Organization

Project Organization

1. Characterize
2. Set Goals
3. Choose Process

Execution plans

4. Execute Process

Experience Factory

1. Characterize
2. Set Goals
3. Choose Process

Execution plans

4. Execute Process

Experience Base

5. Analyze

products, lessons learned, models

project analysis, process modification

data, lessons learned

environment characteristics
tailorable knowledge, consulting

products, lessons learned, models

Project Support

Tailor

Formalize

Disseminate

Generalize

6. Package

Experience Base

©CeBASE
The Goal-Model-Question-Metric (GMQM) Paradigm

• Determine improvement goals
  – Objectives and priorities

• Select models, tailor to organization and goals
  – Assess goal-achievement drivers

• Formulate questions
  – What is our current baseline?
  – What trends and constraints must we consider?
  – What approach (mixed strategy) should we try?
  – How will we assess progress toward goals?

• Define, collect, analyze, manage to metrics
  – Closed-loop feedback cycle
A GMQM Example: TRW, 1985-89

• Goals: Improve productivity and quality
• Models: COCOMO, Value Chain, Cost of Quality
• Questions:
  – Major drivers of productivity and quality?
  – Major trends: user-intensive systems, Ada
  – Approach: Ada process model; architecture; spiral model/risk management
• Metrics: rework effort; effort/SLOC; weighted defects/USLOC
Pareto Analysis of Rework Costs

TRW Project A
373 SPR's

TRW Project B
1005 SPR's

Major Rework Sources:
Off-Nominal Architecture-Breakers
A - Network Failover
B - Extra-Long Messages
Reducing Software Cost-to-Fix: CCPDS-R
- Royce, 1998

— Architecture first
  - Integration during the design phase
  - Demonstration-based evaluation

— Risk Management

— Configuration baseline change metrics:

![Diagram showing project development schedule with change metrics]

- Design Changes
- Implementation Changes
- Maintenance Changes and ECP’s
Effects of the SEL Activities

Continuous Improvement in the SEL

- Decreased Development Defect rates by
  - 75%(87-91)  37%(91-95)
- Reduced Cost by
  - 55%(87-91)  42%(91-95)
- Improved Reuse by
  - 300%(87-91)  8%(91-95)
- Increased Functionality five-fold (76-92)

CSC

officially assessed as CMM level 5 and ISO certified (1998),
starting with SEL organizational elements and activities
Results at USC, 1990’s

- Developed and refined new methods and models
  - WinWin Spiral Model, Anchor Point milestones
  - Model-Based (System) Architecting and Software Engineering (MBASE)
  - Schedule as Independent Variable (SAIV) process
  - COCOMO II, Easy Win Win
    - Commercial transitions
- Successfully applied to digital library projects
  - 17 of 19 projects successfully transitioned within tight schedules
  - Client satisfaction rating average: 4.4/5.0
- Transitioning to commercial software organizations
  - Rational, Xerox, C-Bridge, Media Connex
Proposed Application to DoD Software Intensive Systems

• Identify early adopter for pilot application
• Tailor Experience Factory, GMQM to early adopter’s context
• Operate pilot, train organization in self-application
• Expand to increasingly wider usage, self-sustaining operation