COCOMO II and Extensions

Barry Boehm, USC
COCOMO/SCM 13
October 6, 1998
boehm@sunset.usc.edu
http://sunset.usc.edu

Outline

- COCOMO II overview
  - Objectives and strategy
  - Model specifics
- COCOMO II usage example
- COCOMO II experience and plans
  - Model calibration
  - Usage experience and plans
- COCOMO II extensions
Modeling Implications

Economic analysis of SW-intensive products is critical:
- Business case analyses
- Return on Investment
- Rapid Application Development

New approaches make old models less relevant:
- COTS, Web, concurrent processes

Cost/schedule as independent variable:
- Accurate estimates, tracking are critical

Need metrics-based continuous process improvement for long-range competitiveness

COCOMO II Long Term Vision

System objectives: functionality, portability, quality
Corporate parameters: tools, processes, reuse
Evaluate Corporate SW Improvement Strategies

COCOMO 2.0

Rescope

Cost, Schedule, Effort

OK?

COCOMO II calibration data

OK?

Accumulate COCOMO II calibration data

OK?

Revise Milestones, Plans, Resources

Evaluate COCOMO II calibration data

OK?

No

Yes

Milestone expectations

No

Yes

Rescope

End
Expectations Management: The Brief Success of Accurate Estimators

Well-understood software applications \rightarrow\text{Accurate estimators}

\downarrow

Automation of well-understood parts (GUI builders, COTS, 4GL's, etc.) \rightarrow\text{Higher Productivity, Less accurate estimators}

Software Trends Causing Estimation Challenges

- GUI builders, COTS, 4GL's, reuse, breakage
  - Need to rethink size metrics
- Distributed interactive applications
  - Web-based, object-oriented, event-based
  - Middleware effects
- New process models (evolutionary, incremental, spiral)
  - Phases overlap
  - New labor distributions (vs. Rayleigh curve)
  - Where are cost measurement endpoints?
COCOMO II Model Overview

- COCOMO Baseline Overview
- COCOMO II Objectives
- Coverage of Future Market Sectors
- Hierarchical Sizing Model
- Modes Replaced by Exponent Drivers
- Stronger Reuse/Reengineering Model
- Other Model Improvements
COCOMO Baseline Overview II

- Open interfaces and internals
  - Published in “Software Engineering Economics,” Boehm, 1981
- Numerous implementations, calibrations, extensions
  - Incremental development, Ada, new environment technology
  - Arguably the most frequently-used software cost model worldwide.
- Needs reengineering to address new estimation challenges

COCOMO II Objectives

- Develop a 1990’s-2000’s software cost model
  - Addressing new processes and practices
- Retain COCOMO internal, external openness
- Develop database, tool support for continuous model improvement
- Support closed-loop quantitative project management and process improvement
The future of the software practices marketplace

User programming
(55M performers in US in year 2005)

Application generators
(0.6M)  Application composition
(0.7M)  System integration
(0.7M)  Infrastructure
(0.75M)

COCOMO II Coverage of Future SW Practices Sectors

- User Programming: No need for cost model
- Applications Composition: Use object counts or object points
  - Count (weight) screens, reports, 3GL routines
- System Integration: Development of application generators and infrastructure software
  - Prototyping: Applications composition model
  - Early design: Function Points and/or Source Statements and 7 cost drivers
  - Post-architecture: Source Statements and/or Function Points and 17 cost drivers
- Stronger reuse/reengineering model
Baseline Object Point Estimation Procedure

Step 1: Define Object Points: Identify the number of system, report, and SQL parameter points to comprise the application. Assume the standard definition of these objects in your R&D environment.

Step 2: Identify each object's complexity, modularity, and difficulty complexity levels depending on various characteristic dimensions. Use the following scheme:

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Modularity</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Step 3: Weight the factors in each cell using the following scheme. The weights reflect the relative effort required to implement an expansion of that complexity, modularity, and difficulty level.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Modularity</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Step 4: Summarize Object Points: Add all of the weighted object points to get one total: the Object Point count.

Step 5: Sum up percentage increase in effort required to implement the project. Compare the total Object Points to be implemented (OPC or Object Points) to the baseline of 100.

Process/Anchor Point Examples

Rapid App. Devel.
- Spiral-type
  - Ev.Dev., Spiral
  - LCO, LCA, IOC

Sys Devel
- Spiral-type
  - Waterfall, Spiral-type
  - W*all, IncDev, EvDev, Spiral, Design-to-Cost, etc.
  - LCO, LCA, IOC
Early Design and Post-Arch Model

- Effort:
  \[ PM_{estimated} = AX(Size)^{SF} \times \prod EM \]

- Size
  - KSLOC (Thousands of Source Lines of Code)
  - UFP (Unadjusted Function Points)
  - EKSLOC (Equivalent KSLOC) used for adaptation

- SF: Scale Factors (5)
- EM: Effort Multipliers (7 for ED, 17 for PA)

COCOMO II Model Overview

- COCOMO Baseline Overview
- COCOMO II Objectives
- Coverage of Future Market Sectors
- Hierarchical Sizing Model
- Modes Replaced by Exponent Drivers
- Stronger Reuse/Reengineering Model
- Other Model Improvements
New Scaling Exponent Approach

- Nominal person-months = $A \times (\text{size})^B$
- $B = 0.91 + 0.01 \sum (\text{exponent driver ratings})$
  - $B$ ranges from 0.91 to 1.23
  - 5 drivers; ratings from 0 to 5
- Exponent drivers:
  - Precededness
  - Development flexibility
  - Architecture/risk resolution
  - Team cohesion
  - Process maturity (derived from SEI CMM)

Project Scale Factors

$$PM_{\text{estimated}} = A \times (\text{Size})^{(SF) \times \prod E_M}$$

$$SF = 0.91 + 0.01 \times \sum w_i$$

<table>
<thead>
<tr>
<th>Scale Factors</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC</td>
<td>thoroughly unprecedented</td>
<td>some</td>
<td>well-organized</td>
<td>good</td>
<td>highly</td>
<td>extremely</td>
</tr>
<tr>
<td>FLEX</td>
<td>rigorous</td>
<td>occasional</td>
<td>little</td>
<td>moderate</td>
<td>generally</td>
<td>largely</td>
</tr>
<tr>
<td>RESL</td>
<td>never</td>
<td>very often</td>
<td>often</td>
<td>sometimes</td>
<td>generally</td>
<td>rarely</td>
</tr>
<tr>
<td>TEAM</td>
<td>very difficult</td>
<td>some difficulty</td>
<td>basically</td>
<td>cooperative</td>
<td>highly</td>
<td>cooperative</td>
</tr>
<tr>
<td>PMAT</td>
<td>1</td>
<td>weight is sum of KPA achievement levels</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reuse and Product Line Management

- Challenges
  - Estimate costs of both reusing software and developing software for future reuse
  - Estimate extra effects on schedule (if any)
- Responses
  - New nonlinear reuse model
  - Cost of developing reusable software expanded from Ada COCOMO
  - Gathering schedule data

Nonlinear Reuse Effects

Data on 2954 NASA modules [Selby, 1988]

Amount Modified

Relative cost

Usual Linear Assumption

0.046

0.25 0.5 0.75 1.0

0.55 0.70

Reuse and Reengineering Effects

- Add Assessment & Assimilation increment (AA)
  - Similar to conversion planning increment
- Add software understanding increment (SU)
  - To cover nonlinear software understanding effects
  - Coupled with software unfamiliarity level (UNFM)
  - Apply only if reused software is modified
- Results in revised Equivalent Source Lines of Code (ESLOC)
  - \[ \text{AAF} = 0.4(DM) + 0.3(CM) + 0.3(IM) \]
  - \[ \text{ESLOC} = \text{ASLOC}[\text{AA}+\text{AAF}(1+0.02(SU)(UNFM))], \text{AAF} \leq 0.5 \]
  - \[ \text{ESLOC} = \text{ASLOC}[	ext{AA}+\text{AAF}(SU)(UNFM)], \text{AAF} > 0.5 \]

Software Understanding Rating / Increment

<table>
<thead>
<tr>
<th>Software Understanding Rating</th>
<th>Very Low</th>
<th>Low</th>
<th>Norm</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy &amp; Clarity</td>
<td>Research between programs and requirements, high confidence</td>
<td>Some confidence between programs and requirements, medium confidence</td>
<td>Moderate confidence between programs and requirements</td>
<td>Good confidence between programs and requirements</td>
<td>Very good confidence between programs and requirements</td>
</tr>
<tr>
<td>Self-Dependence</td>
<td>Unpredictable code, documentation unclear, partial understanding</td>
<td>Some non-compatibility and partial understanding</td>
<td>Good code compatibility and understanding, limited documentation, partial understanding</td>
<td>Good code compatibility, detailed documentation, thorough understanding</td>
<td>Very good code compatibility, well-designed, well-documented, well-implemented, comprehensive documentation</td>
</tr>
</tbody>
</table>

AF (Assessment Factor) = 40
Other Major COCOMO II Changes

- Range versus point estimates
- Requirements Volatility replaced by Breakage %
- Multiplicative cost driver changes
  - Product CD's
  - Platform CD's
  - Personnel CD's
  - Project CD's
- Maintenance model and reuse model not identical

---

Post-Architecture EMs - Product:

<table>
<thead>
<tr>
<th>Model (Size)</th>
<th>Variability</th>
<th>Top</th>
<th>Bottom</th>
<th>High</th>
<th>Low</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Design</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Complexity</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Quality</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Documentation</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Integration</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Reusability</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Size</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

---
### Post-Architecture Complexity:

<table>
<thead>
<tr>
<th>Level</th>
<th>Very Low</th>
<th>Low</th>
<th>Normal</th>
<th>High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code/IA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Post-Architecture EMs - Platform:

<table>
<thead>
<tr>
<th>Category</th>
<th>Very Low</th>
<th>Low</th>
<th>Normal</th>
<th>High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical/Time Controlled Device</td>
<td>Sub-Ms/µs</td>
<td>ms/µs</td>
<td>s/µs</td>
<td>ms/µs</td>
<td>s/µs</td>
</tr>
<tr>
<td>Storage Controlled (STC)</td>
<td>Sub-Ms/µs</td>
<td>ms/µs</td>
<td>s/µs</td>
<td>ms/µs</td>
<td>s/µs</td>
</tr>
</tbody>
</table>

Platform: Valiant (P0x0)

- Power change: 35% (2.5kW, 5kW)
- Power draw: 5kW, 10kW, 20kW
## Post-Architecture Ems - Personnel:

<table>
<thead>
<tr>
<th>Position</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyst Capacity (ACP)</td>
<td>10%</td>
<td>50%</td>
<td>75%</td>
<td>85%</td>
<td>90%</td>
<td>95%</td>
</tr>
<tr>
<td>Programmer Capability (PCC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
<td>50%</td>
<td>60%</td>
<td>70%</td>
</tr>
<tr>
<td>Application Experience (AE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 2 months</td>
<td>1 year</td>
<td>2 years</td>
<td>3 years</td>
<td>4 years</td>
<td>5 years</td>
<td>6 years</td>
</tr>
<tr>
<td>Platform Experience (PX)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 1 months</td>
<td>1 year</td>
<td>2 years</td>
<td>3 years</td>
<td>4 years</td>
<td>5 years</td>
<td>6 years</td>
</tr>
<tr>
<td>Language and Tool Experience (LTE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 2 months</td>
<td>1 year</td>
<td>2 years</td>
<td>3 years</td>
<td>4 years</td>
<td>5 years</td>
<td>6 years</td>
</tr>
</tbody>
</table>

## Post-Architecture EMs - Project:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of Software Tools (TST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual development (MDP)</td>
<td>International</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source code quality assurance</td>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
<td>50%</td>
<td>60%</td>
</tr>
<tr>
<td>Post-Architecture (PA)</td>
<td>70%</td>
<td>80%</td>
<td>85%</td>
<td>90%</td>
<td>95%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Early Design vs. Post-Arch EMs:

Other Model Refinements

- Initial Schedule Estimation

\[ TDEV = 5.67 \times \frac{0.28 + 0.2 \times (B - 0.91)}{100} \times \frac{SCED}{10} \]

where \( FM \) = estimated person months excluding Schedule multiplier effects

- Output Ranges

<table>
<thead>
<tr>
<th>Range</th>
<th>Optimistic Estimate</th>
<th>Pessimistic Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Composition</td>
<td>3.50 E</td>
<td>2.5 E</td>
</tr>
<tr>
<td>Early Design</td>
<td>3.37 E</td>
<td>1.5 E</td>
</tr>
<tr>
<td>Post-Architecture</td>
<td>3.40 E</td>
<td>1.20 E</td>
</tr>
</tbody>
</table>

- 80% confidence limits: 10% of time each below Optimistic, above Pessimistic
- Reflect sources of uncertainty in model inputs
COCOMO Model Comparisons

<table>
<thead>
<tr>
<th>Model</th>
<th>Objectives and strategy</th>
<th>Model specifics</th>
<th>COCOMO II usage example</th>
<th>COCOMO II experience and plans</th>
<th>Model calibration</th>
<th>Usage experience and plans</th>
<th>Model-Based Architecting and Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCOMO II</td>
<td>- Objectives and strategy - Model specifics</td>
<td>- COCOMO II usage example - COCOMO II experience and plans - Model calibration - Usage experience and plans - Model-Based Architecting and Software Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outline

- COCOMO II overview
  - Objectives and strategy
  - Model specifics
- COCOMO II usage example
- COCOMO II experience and plans
  - Model calibration
  - Usage experience and plans
- Model-Based Architecting and Software Engineering
1. LSDL Architecture Summary

Socal University (SCU) has three major campuses, each of which has a main library which will provide LSDL services. Each of three campuses operates a server running a COTS client-server library services package. The COTS package provides Internet capabilities enabling a client at any campus to deal with any of three servers.

The SCU Library, Computer Sciences Department, and Computing Services Operation have been funded to develop an experimental Selective Dissemination of Information (SDI) system to provide SCU users with information about new library acquisitions of interest. It will do this by comparing attributes of new library acquisitions with interest profiles provided by library users. The grant provides $850K to develop the system.

The SDI software will acquire data on recent library acquisitions, compare it with library user interest profiles, generate user notices, and support query, browsing, and request functions for the new acquisitions. It will have a number of subsidiary functions for user interest profile management, access control, user interface functions, and usage monitoring. It will communicate with users via the COTS client-server package.

2. Candidate SDI Software Functions and Cost Driver Ratings

Candidate Software Functions

- User Interest Profile Management (5 KDSI). User interest profile creation, deletion, update, query, etc.
- Access Control:
  - Basic (4 KDSI). Basic password access control.
  - Extended (8 KDSI). Basic plus authentication, authorization, instruction deletion.
  - Rigorous (15 KDSI). Extended plus formal specification and verification.
- Acquisition Data Handling and Profile Checking: acquisition record and file access, validity checking, logging, storage, etc. Determination of matches between acquisition and interest profiles.
  - Basic Service (3 KDSI). Simple matching; standard library categories.
  - Extended Service (6 KDSI). Conditional matching; standard plus locally-defined categories.
- User Interface and Services
  - Basic Service (7 KDSI). Acquisition-match query, browse, and request functions. Basic help and profile management functions.
  - Extended Service (12 KDSI). Basic Service plus tutorial functions, interoperability with COTS user interface.
• Usage Monitoring (4 KDSI). Recording and summarizing system usage by time of day/week/year, by use attributes, by acquisition attributes.

• Trend Analysis (3 KDSI). Extrapolation of rates of increase, decrease, or cyclical trends in services.

• Library network access (6 KDSI). Extension of acquisition notification to cover acquisitions from a regional network of 10 additional universities.

• COTS integration. Software needed to integrate LSD1 functions with COTS library information functions:
  - 5 KDSI for processor X;
  - 8 KDSI for processor Y.

COCOMO II software development and cost drivers
Some of the COCOMO II cost drivers are associated with hardware options to be specified in the next section. Below are the ratings determined for the reminder of the LSD1 system cost drivers. Most of them are the same for all of the candidate software functions; any differences are identified on the following page.

<table>
<thead>
<tr>
<th>Effort Multipliers</th>
<th>RELY Nominal, except for Access Control, which is High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA High</td>
</tr>
<tr>
<td></td>
<td>CPLX Nominal, except for Access Control (High) and COTS Integration (High)</td>
</tr>
<tr>
<td></td>
<td>RUSE Nominal</td>
</tr>
<tr>
<td></td>
<td>DOCU Nominal</td>
</tr>
<tr>
<td></td>
<td>STOR Nominal</td>
</tr>
<tr>
<td></td>
<td>FVOL Low</td>
</tr>
<tr>
<td></td>
<td>ACAP Nominal</td>
</tr>
<tr>
<td></td>
<td>PCAP Nominal</td>
</tr>
<tr>
<td></td>
<td>AEXP Nominal</td>
</tr>
<tr>
<td></td>
<td>PCON Nominal</td>
</tr>
<tr>
<td></td>
<td>LTEX High</td>
</tr>
<tr>
<td></td>
<td>SITE Very High</td>
</tr>
<tr>
<td></td>
<td>SCED Nominal</td>
</tr>
<tr>
<td></td>
<td>Scale Factors</td>
</tr>
<tr>
<td></td>
<td>PREC Nominal</td>
</tr>
<tr>
<td></td>
<td>FLEX High</td>
</tr>
<tr>
<td></td>
<td>RESL Nominal</td>
</tr>
<tr>
<td></td>
<td>TEAM Very High</td>
</tr>
<tr>
<td></td>
<td>PMAT Nominal</td>
</tr>
</tbody>
</table>

10/6/96
3. Hardware Options
There are two hardware options for SDI functions. Processor X is a more mature but slower processor, with better tool support. Processor Y is a fast, new processor with a lower level of tool support and virtual machine experience. A summary of their comparative cost/performance/risk feature is as follows:

<table>
<thead>
<tr>
<th></th>
<th>Processor X</th>
<th>Processor Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCOMO TIME rating</td>
<td>High</td>
<td>Nominal</td>
</tr>
<tr>
<td>COCOMO PEXP rating</td>
<td>Nominal</td>
<td>Low</td>
</tr>
<tr>
<td>COCOMO TOOL rating</td>
<td>High</td>
<td>Nominal</td>
</tr>
<tr>
<td>Nominal response time</td>
<td>5 sec</td>
<td>1 sec</td>
</tr>
<tr>
<td>System development risk</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Hardware cost</td>
<td>$12K</td>
<td>$20K</td>
</tr>
</tbody>
</table>

Outline

- COCOMO II overview
  - Objectives and strategy
  - Model specifics
- COCOMO II usage example
- COCOMO II experience and plans
  - Model calibration
  - Usage experience and plans
- Model-Based Architecting and Software Engineering
Model Calibration Status

- Three models comprise COCOMO II:
  - Applications Composition
  - Early Design
  - Post-Architecture
  - Post-Architecture model calibrated
  - Early Design model derived from Post-Architecture
  - Applications Composition: need data

Calibration Process

- Began with expert-determined a-priori model parameters
- Collected Data
- Identified and consolidated highly correlated model parameters
- Statistically determined estimates of consolidated model parameters from data
- Used data-determined coefficients to adjust a-priori model parameters
- Experimented with weighting factors
Expanded Post-Architecture Model:

- Distribute the Scale Factors
- 21 predictor variables: 15 Effort Multipliers + 5 Scale Factors + (Size)\(^{1.01}\)

\[
PM = A \times (\text{Size})^{1.01} \times (\text{Size})^{SF_1} \times (\text{Size})^{SF_2} \times \ldots \times EM_1 \times \ldots \times EM_r
\]

Log Transformed Model:

- Regression analysis will derive the coefficients, \(b_i\), for each factor

\[
\ln(PM_{\text{fin}}) - \ln(\text{Size})^{1.01} \times \ln(A) + \sum_{i=1}^{SF} \ln(SF_i) + \sum_{j=1}^{EM} \ln(EM_j)
\]

RUSE Effort Multiplier:

- Example of the effect of a negative coefficient
Distribution of RUSE:

Evolving Model Values: 1997

Bayesian

100% Data Driven

100% Expert Driven

Number of projects used in calibration

Bayes' Theorem

\[ g(\beta | y) = \frac{f(y | \beta) g(\beta)}{f(y)} \]

posterior information \( \alpha \) sample information \( \times \) prior information

A-Priori Information + Sampling Data = A-Posteriori Model
COCOMO II Calibration Approaches

Literature, Behavioral Analysis (Steps 1-3)

Language and Tool Experience (LTEX)
Results of Delphi (Step 4)

Language Tool and Experience (LTEX)

Results of Sampling Data (Step 5)

Language and Tool Experience (LTEX)
Results of Bayesian Update: Using Prior and Sampling Information (Step 6)

Language and Tool Experience (LTEX)

Bayesian Analysis of Process Maturity Productivity Range (100 KSLOC)

(experts vs. beginners)
Distribution of Size: 1998 database

Distribution of Effort: 1998 database
Distribution of Schedule: 1998 database

Provisional COCOMO II, 1998 Productivity Ranges
### Accuracy Results:
Forecast accuracy measured with relative error:

\[ \text{RE} = \frac{|X_{\text{act}} - X_{\text{pred}}|}{X_{\text{act}}} \]

<table>
<thead>
<tr>
<th>Effort Prediction</th>
<th>Before Stratification by Organization</th>
<th>After Stratification by Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pred.(20)</td>
<td>46%</td>
<td>53%</td>
</tr>
<tr>
<td>Pred.(25)</td>
<td>49%</td>
<td>68%</td>
</tr>
<tr>
<td>Pred.(30)</td>
<td>52%</td>
<td>75%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pred (20)</td>
<td>48%</td>
<td>54%</td>
<td>52%</td>
<td>60%</td>
</tr>
<tr>
<td>Pred (25)</td>
<td>51%</td>
<td>65%</td>
<td>61%</td>
<td>71%</td>
</tr>
<tr>
<td>Pred (30)</td>
<td>61%</td>
<td>72%</td>
<td>62%</td>
<td>81%</td>
</tr>
</tbody>
</table>

### COCOMO II. 1999 Parameter Values

**Effort Coefficient = 2.94; Scale Factor Base = 0.91**

**Schedule Coefficient = 3.67; Scale Factor base = 0.28**

#### Scale Factors

<table>
<thead>
<tr>
<th>Wig</th>
<th>Very Low</th>
<th>Low</th>
<th>Normal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedent Maturity</td>
<td>6.20</td>
<td>4.96</td>
<td>3.72</td>
<td>2.48</td>
<td>1.24</td>
<td>0.00</td>
</tr>
<tr>
<td>Development Flexibility</td>
<td>5.07</td>
<td>4.05</td>
<td>3.04</td>
<td>2.03</td>
<td>1.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Architectural Risk Resolution</td>
<td>7.07</td>
<td>5.65</td>
<td>4.24</td>
<td>2.83</td>
<td>1.41</td>
<td>0.00</td>
</tr>
<tr>
<td>Team Cohesion</td>
<td>5.48</td>
<td>4.38</td>
<td>3.29</td>
<td>2.15</td>
<td>1.10</td>
<td>0.00</td>
</tr>
<tr>
<td>Process Maturity</td>
<td>7.80</td>
<td>6.24</td>
<td>4.68</td>
<td>3.12</td>
<td>1.56</td>
<td>0.00</td>
</tr>
</tbody>
</table>

105/98 57

106/98 58
Values-Effort Multipliers

USC COCOMO II. 1999 Parameter Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Very Low</th>
<th>Low</th>
<th>Normal</th>
<th>High</th>
<th>Very High</th>
<th>Extreme High</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>1.00</td>
<td>1.00</td>
<td>1.10</td>
<td>1.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>0.90</td>
<td>1.00</td>
<td>1.10</td>
<td>1.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRTA</td>
<td>0.70</td>
<td>0.80</td>
<td>1.10</td>
<td>1.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIVE</td>
<td>0.70</td>
<td>0.90</td>
<td>1.10</td>
<td>1.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPUT</td>
<td>0.80</td>
<td>1.00</td>
<td>1.10</td>
<td>1.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBK</td>
<td>1.00</td>
<td>1.10</td>
<td>1.50</td>
<td>1.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STER</td>
<td>1.00</td>
<td>1.05</td>
<td>1.10</td>
<td>1.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FVOL</td>
<td>0.60</td>
<td>0.70</td>
<td>1.00</td>
<td>1.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACFAP</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAP</td>
<td>1.50</td>
<td>1.50</td>
<td>1.50</td>
<td>1.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCON</td>
<td>1.20</td>
<td>1.20</td>
<td>1.50</td>
<td>1.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEXP</td>
<td>1.20</td>
<td>1.20</td>
<td>1.50</td>
<td>1.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FESP</td>
<td>1.00</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIPS</td>
<td>1.00</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WML</td>
<td>1.00</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFR</td>
<td>1.00</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCRD</td>
<td>1.40</td>
<td>1.40</td>
<td>1.60</td>
<td>1.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COCOMO II. 1999 Parameter Values

USC COCOMO II.1998

- Project data entry and calibration
- Incremental rating scales
- New Bayesian parameter values
- Windows, Unix/ Motif, Java versions
- Feature and user interface cleanup
- Export to Excel (import underway)
- Extensive on-line help
  - User Manual, Model Definition Manual
- Coordination with commercial COCOMO II products
- Related tools: Code Count, Bug tracking, Spreadsheet versions of Early Design, Post-Architecture, COSEMO, CORADMQ, COPROMO
Usage Experience to Date

- Most experience with Post-Architecture model
  - Basic form works pretty well
  - Data conditioning harder
- Equating reuse and maintenance models didn’t work
  - SU increment applies only to part of legacy software
- Function points weak for sizing reuse, maintenance
- Added capabilities needed

Information Sources:

- Phone: (213) 740-6470
- Email: cocomo-info@sunet.usc.edu
- Web site: http://sunet.usc.edu/COCOMOII/Cocomo.html
  - Affiliate Prospectus
  - Model Definition Manual (ver. 1.4)
  - Data Collection Form (ver. 1.6)
  - Java COCOMO
  - Little Expert COCOMO Calculator
Outline

- COCOMO II Overview
- Overview of Extensions
  - 1997 Affiliate Priorities
  - COTS Integration (COCOTS)
  - Quality: Delivered Defect Density (COQUALMO)
  - Stage Distributions (COSSEMO)
  - Rapid Application Development Schedule (CORADMO)
  - Productivity Improvement (COPROMO)
  - Tool Effects
  - UML-Based Sizing
- Next Steps

1997 Affiliate Priorities

21 • Improve accuracy of COCOMO II Model
15 • Cost/schedule/quality tradeoffs
12 • Sizing improvements
10 • COTS integration costs
 7 • Activity Distribution
 3 • Life cycle tradeoff models
USC-CSE Modeling Methodology

Status of Models

<table>
<thead>
<tr>
<th>Literature</th>
<th>Behavior</th>
<th>Signif. Variables</th>
<th>Delphi</th>
<th>Data, Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCOMO II</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>161</td>
</tr>
<tr>
<td>CoCoTS</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-10</td>
</tr>
<tr>
<td>COQUALMO</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>Defects in</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defects out</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>COSSEMO</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CORADNO</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**COCOMO vs. COCOTS Cost Sources**

- COCOMO
  - Local Previews
  - Lab (FDG)
  - COCOMO Cost Estimation

- COCOTS
  - Application Code Development and Testing
  - Software Development Effort, Cost, and Schedule Estimate

**Integrated COQUALMO**

- COCOMO
  - Software Size Estimate
  - Software Product, Process, Computer, and Personnel Attributes
  - Defect Removal Capability Levels

- COQUALMO
  - Defect Removal Model

- Software development effort, cost, and schedule estimates
- Number of residual defects
- Defect density per unit of size
Need to Improve Classic Schedule Model

Months ~ $3^{\frac{3}{2}}$ Person-Months

- Overestimates for small RAD projects
- No reflection of RAD Opportunity Tree tradeoffs
  - Reuse, VHLL's (RVHL)
  - Business Process Reengineering (BPRS)
  - Collaboration Technology (CLAB)
  - Architecture, Risk Resolution (RESL)
  - Preposition Assets (PPOS)

RAD Opportunity Tree

<table>
<thead>
<tr>
<th>Eliminating Tasks</th>
<th>Reducing Time Per Task</th>
<th>Reducing Risks of Single-Point Failures</th>
<th>Reducing Backmasking</th>
<th>Activity Network Streamlining</th>
<th>Increasing Effective Workweek</th>
<th>Better People and Incentives</th>
<th>Transition to Learning Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business process reengineering - BPRS</td>
<td>Applications generation - RHVL</td>
<td>Reducing failures - RESL</td>
<td>Early error elimination - RESL</td>
<td>Minimizing task dependencies - BPRS</td>
<td>Nightly builds, testing - PPOS</td>
<td>- constraint</td>
<td>-</td>
</tr>
<tr>
<td>Reuse assets - RHVL</td>
<td>Design-to-schedule - 0</td>
<td>Reducing their effects - RESL</td>
<td>Process anchor points - RESL</td>
<td>Avoiding high-level, late-stage - BPRS</td>
<td>Weekend workforces - PPOS</td>
<td>- constraint</td>
<td>-</td>
</tr>
<tr>
<td>Applications generation - RHVL</td>
<td>Work streamlining (MO: 0)</td>
<td>Increasing parallelism - RESL</td>
<td>Improving process maturity - 0</td>
<td>Reducing task variance - BPRS</td>
<td>Reducing tasks from critical path - BPRS</td>
<td>- constraint</td>
<td>-</td>
</tr>
<tr>
<td>Design-to-schedule - 0</td>
<td>Incrementing parallelism - RESL</td>
<td>Reducing tasks - RESL</td>
<td>Collaboration Technology - CLAB</td>
<td>Removing tasks from critical path - BPRS</td>
<td>24x7 development - PPOS</td>
<td>- constraint</td>
<td>-</td>
</tr>
</tbody>
</table>

10/5/98 69

0 covered by classic cube root model 70
COCOMO II RAD Extension (CORADMO)

COCOMO II cost drivers (except SCED),
Language, level, experience...

Baseline effort, schedule
RAD effort, schedule by stage

RAD Extension

COCOMO II Baseline effort, schedule by stage

Stage Distributions

COCOMO II Effort Schedule

COSSEMO Stage Distribution Model: Early Design/ Post Architecture

<table>
<thead>
<tr>
<th>LCO</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Total EAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort %</td>
<td>14</td>
<td>28</td>
<td>72</td>
<td>100</td>
</tr>
<tr>
<td>Schedule %</td>
<td>40</td>
<td>40</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>P / M = P</td>
<td>0.35</td>
<td>0.7</td>
<td>1.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ps / Ms-P</th>
<th>32K</th>
<th>512K</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.414.8 = 3.5</td>
<td>36.1/13.7 = 2.6</td>
<td>720/3.7 = 190</td>
</tr>
</tbody>
</table>

RVHL
BPRS
CLAB
RESL
PPOS

Inception +14 28 72 100
Elaboration
Construction 100

Effort 16.4/14.8 = 3.5
Schedule 36.1/13.7 = 2.6
P / M = P 720/3.7 = 190
Total EAC 3580/54.3 = 75
COSSEMO Duration Calculation

![Graph showing duration calculation](image)

CORADMO: Reuse and VHLL’s (RVHL)

- Standard 3GL module reuse: no adjustment
- Schedule compression in Inception and Elaboration stages due to faster prototyping, option exploration
  - effect depends on experience in doing this
  - compression/expansion doesn’t alter staff level

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Rapid Prototyping Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multipliers</td>
</tr>
<tr>
<td>Inception</td>
<td>1.05</td>
</tr>
<tr>
<td>Elaboration</td>
<td>1.03</td>
</tr>
<tr>
<td>Construction</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Development Process Reengineering and Streamlining (DPRS)

- Detailed rating scale provided
- Gains depend on current level of bureaucracy - compression/expansion doesn't alter staff level

<table>
<thead>
<tr>
<th>Schedule Multipliers</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL - Heavily bureaucratic</td>
<td>1.30</td>
<td>1.30</td>
<td>1.20</td>
</tr>
<tr>
<td>L - Bureaucratic</td>
<td>1.12</td>
<td>1.08</td>
<td>1.08</td>
</tr>
<tr>
<td>N - Basic good business practices</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>H - Partially streamlined</td>
<td>.96</td>
<td>.95</td>
<td>.95</td>
</tr>
<tr>
<td>VH - Fully streamlined</td>
<td>.92</td>
<td>.95</td>
<td>.95</td>
</tr>
</tbody>
</table>

DPRS Rating Scale

<table>
<thead>
<tr>
<th>Number of approvals required per task</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken pre approval</td>
<td>Excessive</td>
<td>Occasional</td>
<td>Rare</td>
<td>Reduced</td>
<td>Activity Reduced</td>
</tr>
<tr>
<td>Reduced task dependencies, critical path tasks</td>
<td>Sure</td>
<td>Little</td>
<td>Mature Tasks, Adopted</td>
<td>Advanced</td>
<td>Strong Streamlining</td>
</tr>
<tr>
<td>Followup to expedite task completion</td>
<td>None</td>
<td>Little</td>
<td>Mature Tasks, Adopted</td>
<td>Advanced</td>
<td>Strong Streamlining</td>
</tr>
<tr>
<td>Process measurement &amp; streamlining</td>
<td>None</td>
<td>Little</td>
<td>Mature Tasks, Adopted</td>
<td>Advanced</td>
<td>Strong Streamlining</td>
</tr>
</tbody>
</table>
Collaboration Technology (CLAB)

- Basic rating scale and multipliers provided
- Detailed rating scale complex, TBD
  - Effects dependent on TEAM, SITE rating
  - Needs integration with TOOL rating
  - Collaboration tool maturity, experience, scope (domain, negotiation, option-analysis tool support)
- Same effect on effort; same staff level

<table>
<thead>
<tr>
<th>Schedule and Effort Multipliers</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>EH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception</td>
<td>1.31</td>
<td>1.10</td>
<td>1.00</td>
<td>0.93</td>
<td>0.86</td>
<td>0.80</td>
</tr>
<tr>
<td>Elaboration</td>
<td>1.15</td>
<td>1.07</td>
<td>1.00</td>
<td>0.95</td>
<td>0.90</td>
<td>0.86</td>
</tr>
<tr>
<td>Construction</td>
<td>1.10</td>
<td>1.05</td>
<td>1.00</td>
<td>0.98</td>
<td>0.95</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Architecture / Risk Resolution (RESL)

- Same as CCOMO II RESL rating scale
- Enables parallel construction

<table>
<thead>
<tr>
<th>Schedule Multipliers (Effort Unchanged)</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>EH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Elaboration</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Construction</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>.91</td>
<td>.83</td>
<td>.75</td>
</tr>
</tbody>
</table>
Prepositioning Assets (PPOS)

- Degree to which assets are pre-tailored to project and furnished to project for use on demand
- People skills and teambuilding
- Processes and tools
- Architecture and componentry

### Table: Prepositioning Multiples

<table>
<thead>
<tr>
<th>Rating</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>EH</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-hot collection, Some incompatibilities</td>
<td>1.10/1.05x1.0</td>
<td>1.0</td>
<td>1.06/1.07x1.02</td>
<td>1.13/1.06x1.12</td>
<td>1.2/1.09x1.46</td>
</tr>
<tr>
<td>Basic project legacy, re-building</td>
<td>1.10/1.00x1.0</td>
<td>1.0</td>
<td>1.06/1.12x1.12</td>
<td>1.13/1.25x1.25</td>
<td>1.2/1.41x1.41</td>
</tr>
<tr>
<td>Some prepositioning &amp; tailoring</td>
<td>1.05/1.05x1.0</td>
<td>1.0</td>
<td>1.06/1.07x1.07</td>
<td>1.13/1.25x1.25</td>
<td>1.2/1.41x1.41</td>
</tr>
</tbody>
</table>

### People Capabilities as Constraint

- Use Early Design Personnel Capability Rating (ACAP, PCAP, PCON, Merge) as constraint
- RAD driver ratings can't be higher than personnel capability rating
### Application Development Example - I

- Use best possible schedule compression rating

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM / M = P</td>
<td>0.92 / 0.92 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
<td>1.00 / 1.00 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
</tr>
<tr>
<td>PM / M = P</td>
<td>0.92 / 0.92 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
<td>1.00 / 1.00 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
</tr>
<tr>
<td>PM / M = P</td>
<td>0.92 / 0.92 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
<td>1.00 / 1.00 = 1.00</td>
<td>0.95 / 0.95 = 1.00</td>
</tr>
</tbody>
</table>

10/5/98 81

---

### Application Development Example - I

**32 KSLOC Project (PM / M = P)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>16/0/4.9 = 3.3</td>
<td>3.6/4.9 = 7.5</td>
<td>8.6/7.2 = 13</td>
<td>123/12 = 10</td>
</tr>
<tr>
<td>RAD</td>
<td>13/6/2.8 = 4.8</td>
<td>31/2/3.2 = 9.8</td>
<td>91/6/4 = 23</td>
<td>123/7.2 = 17</td>
</tr>
</tbody>
</table>

**512 KSLOC Project**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>36/1/13.7 = 26</td>
<td>72/2/13.7 = 53</td>
<td>198/8/20.6 = 90</td>
<td>258/3/24.3 = 75</td>
</tr>
<tr>
<td>TT</td>
<td>.81/1/5.6 = 1.41</td>
<td>9/9/5.6 = 1.41</td>
<td>1.06/5.6 = 1.88</td>
<td>1.20/5 = 1.70</td>
</tr>
<tr>
<td>RAD</td>
<td>29/2/7.95 = 97</td>
<td>67/1/9 = 74</td>
<td>196/9/11.5 = 171</td>
<td>283/2/28.2 = 103</td>
</tr>
</tbody>
</table>

10/5/98 81

41
Productivity improvement Model

- Use COCOMO II model and extensions as assessment framework
  - Well-calibrated to 161 projects for effort, schedule
  - Subset of 106 1990's projects for current-practice baseline
  - Extensions for Rapid Application Development formulated
- Determine likely near-term (2006) and longer-term (2013) impact of technologies on model parameter settings
- Use these in models to assess impact of technologies on cost and schedule
  - Effort used as a proxy for cost
- Useful for corporate benchmarking and improvement strategy evaluation

COCOMO II.1998 Productivity Ranges and Current Practice

[Graph showing productivity ranges with average multiplier indicated for 1990's projects]
COCOMO II Tool Refinement

- No Comparison of the Same Kind of Tools
- No Clear Definition of Tools
- No Considerations of Interactions with Other Factors

Basis of Tool Rating Scale

<table>
<thead>
<tr>
<th>Basis of Tool Rating Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth of Process Support</td>
</tr>
<tr>
<td>Specification, Analysis,</td>
</tr>
<tr>
<td>Design, Programming,</td>
</tr>
<tr>
<td>Test, CM, QA, Management,</td>
</tr>
<tr>
<td>CMM Tool maturity and</td>
</tr>
<tr>
<td>support</td>
</tr>
<tr>
<td>Degree of Tool Integration</td>
</tr>
</tbody>
</table>

New Tool Rating Scale
UML-Based Sizing

- In early exploration stages
- Exploring hierarchical approach

Number of properties of use cases

Number of properties of intermediate views
- classes, objects, components, sequence/collaboration diagrams, state transition diagrams

Number of properties of source code statements

UML Sizing Challenges

- UML-view usage conventions
- View overlaps and double counting
- Size vs. complexity measures
- Counting attributes and relationships
- Data collection
Next Steps

- Discuss status and prospects in Workshop
  - Delphi exercises
- Determine data availability, priorities
- Use to prioritize projects
  - Committed to COCOTS, COQUALMO, TOOL effects

List of Acronyms

- COCOMO: COnstructive COst MOdel
- COQUALMO: COnstructive QUALity MOdel
- COCOTS: COnstructive COTS Integration Model
- COSSEMO: COnstructive Staged Schedule Effort Model
- CORADMO: COnstructive RAD MOdel
- COPROMO: COnstructive PROductivity Improvement MOdel
- COTS: Commerical -Off-The-Shelf Software
- IOC: Initial Operational Capability milestone
- LCA: Life Cycle Architecture milestone
- LCO: Life Cycle Objectives milestone
- RAD: Rapid Application Development
- UML: Unified Modeling Language