COCOMO II Model

Brad Clark
CSE Research Associate
15th COCOMO/SCM Forum
October 22, 1998
Brad@Software-Metrics.com
COCOMO II Model Overview

- COCOMO II Overview
- Sizing the Application
- Estimating Effort
- Estimating Schedule
- Understanding model workings
- Estimating Software Maintenance
Early Design and Post-Arch Models

- Nominal-Schedule Estimated Effort ($PM_{NS}$):
  (excludes Required Development Schedule cost driver)

$$PM_{NS} = A \times (\text{Size})^E \times \prod_{i=1}^{n} EM_i$$

where $E = B + 0.01 \times \sum_{j=1}^{5} SF_j$

- $A = 2.94$  \quad $B = 0.91$
- Size (thousands of lines of code, KSLOC, or function points)
- $EM$: Effort Multipliers (6 for ED, 16 for PA)
- $SF$: Scale Factors (5 for both models)
Early Design and Post-Arch Models

• Nominal-Schedule Estimated Duration ($TDEV_{NS}$):
  (excludes Required Development Schedule cost driver)

\[
TDEV_{NS} = C \times (PM_{NS})^F
\]

where \( F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j \)

- \( C = 3.67 \)
- \( PM_{NS} = \) from nominal-schedule effort estimation
- \( D = 0.28 \)
- \( SF: \) Scale Factors (5 for both models)
Post-Arch Model Example

• Suppose we had a 100 KSLOC, average project
  – Sum EM’s = 1.0 (all nominal ratings)
  – Sum SF’s = 24 (mostly low ratings)

• $PM_{NS}=2.94(100)^{(0.91+0.01*24)}=586.61$ person months
• $TDEV_{NS}=3.67(586.6)^{(0.28+0.2*0.01*24)}=29.7$ months
• Average number of staff = $PM_{NS}/TDEV_{NS} = 19.75$ people
COCOMO II Model Overview

• COCOMO II Overview
• Sizing the Application
• Estimating Effort
• Estimating Schedule
• Understanding model workings
• Estimating Software Maintenance
KSLOC: Thousands of Source Lines of Code
CR: Conversion Ratios
REVL: Requirements Evolution
AAM: Adaptation Adjustment Modifiers
Function Point Definition

- External User
  - External Input
  - External Output
  - External Inquiry

- Internal Logical Files
- Application Boundary
- External Interface Files
- External Logical Files
- Another Application
# Some Function Point Conversion Ratios

<table>
<thead>
<tr>
<th>Language</th>
<th>SLOC / UFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada 83</td>
<td>71</td>
</tr>
<tr>
<td>Ada 95</td>
<td>49</td>
</tr>
<tr>
<td>AI Shell</td>
<td>49</td>
</tr>
<tr>
<td>APL</td>
<td>32</td>
</tr>
<tr>
<td>Assembly - Basic</td>
<td>320</td>
</tr>
<tr>
<td>Assembly - Macro</td>
<td>213</td>
</tr>
<tr>
<td>Basic – ANSI</td>
<td>64</td>
</tr>
<tr>
<td>Basic – Compiled</td>
<td>91</td>
</tr>
<tr>
<td>Basic – Visual</td>
<td>32</td>
</tr>
<tr>
<td>C</td>
<td>128</td>
</tr>
<tr>
<td>C++</td>
<td>55</td>
</tr>
<tr>
<td>Cobol (ANSI 85)</td>
<td>91</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
SLOC Definition Considerations

• Whether to include or exclude
  – executable and/or non-executable code statements
  – code produced by programming, copying without change, automatic generation, and/or translation
  – newly developed code and/or previously existing code
  – product-only statements or also include support code
  – counts of delivered and/or non-delivered code
  – counts of operative code or include dead code
  – replicated code

• When does the code get counted
  – at estimation, at design, at coding, at unit testing, at integration, at test readiness review, at system test complete
Using Preexisting Code

• Reused Code
  – Preexisting code that is treated as a black-box and plugged into the product

• Adapted Code
  – Preexisting code that is treated as white-box and is modified for use with the product

• The size of reused and adapted code is adjusted to be its equivalent in new code using Adaptation Adjustment Modifiers (AAM)
  – Based on additional effort it takes to modify the code for inclusion in the product
  – Adaptation with function points and source lines of code is the same for Early Design and Post-Arch Models
Non-Linear Reuse Size Model

• This is the reuse size model and is its effects are shown on the next slide
  - AT is the percentage of code that is reengineered by automatic translation
  - The other factors are explained next

Equivalent KSLOC = Adapted KSLOC \times \left( 1 - \frac{AT}{100} \right) \times AAM

\[
AAM = \begin{cases} 
\frac{[AA + AAF(1 + (0.02 \times SU \times UNFM))]}{100}, & \text{for } AAF \leq 50 \\
\frac{[AA + AAF + (SU \times UNFM)]}{100}, & \text{for } AAF > 50
\end{cases}
\]

\[
AAF = (0.4 \times DM) + (0.3 \times CM) + (0.3 \times IM)
\]
Nonlinear Reuse Effects

AAM Worst Case:
- AAF = 0.5
- AA = 8
- SU = 50
- UNFM = 1

AAM Best Case:
- AAF = 0.5
- AA = 0
- SU = 10
- UNFM = 0

Data on 2954 NASA modules [Selby, 1988]

Selby data summary
Adaptation Adjustment Modifiers

• **AAF**: Adaptation Adjustment Factor (COCOMO 81)
  – Percent Design and Code Modified (DM and CM)
  – Percent Integration Required for Adapted Software (IM)

• **AA**: Assessment and Assimilation increment (0 - 8)
  – Determine whether a reused software module is appropriate to the application and to integrate it description into the overall product description

• **SU**: Software Understanding increment (50 - 10)
  – To cover nonlinear software understanding effects
  – Coupled with software unfamiliarity level (UNFM)
  – Apply only if reused software is modified

• **UNFM**: Programmer Unfamiliarity
  – Handles situations where the programmers are familiar with the software to be adapted.
### SU Rating / Increment Example

<table>
<thead>
<tr>
<th>Structure</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very low cohesion, high coupling, spaghetti</td>
<td>Moderately low cohesion,</td>
<td>Reasonably well-structured; some weak</td>
<td>High cohesion, low coupling.</td>
<td>Strong modularity, information hiding in data / control structures.</td>
</tr>
<tr>
<td></td>
<td>code.</td>
<td>high coupling.</td>
<td>areas.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Clarity</td>
<td>No match between program and application</td>
<td>Some correlation between</td>
<td>Moderate correlation between program and</td>
<td>Good correlation between program and</td>
<td>Clear match between program and application</td>
</tr>
<tr>
<td></td>
<td>world-views.</td>
<td>program and application.</td>
<td>application.</td>
<td>application.</td>
<td>world-views.</td>
</tr>
<tr>
<td>Self-Descriptiveness</td>
<td>Obscure code; documentation missing,</td>
<td>Some code commentary and</td>
<td>Moderate level of code commentary,</td>
<td>Good code commentary and headers; useful</td>
<td>Self-descriptive code; documentation</td>
</tr>
<tr>
<td></td>
<td>obscure or obsolete</td>
<td>headers; some useful</td>
<td>headers, documentation.</td>
<td>documentation; some weak areas.</td>
<td>up-to-date, well-organized, with design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>documentation.</td>
<td></td>
<td></td>
<td>rationale.</td>
</tr>
<tr>
<td>SU Increment to ESLOC</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
COCOMO II Model Overview

• COCOMO II Overview
• Sizing the Application
• Estimating Effort
• Estimating Schedule
• Understanding model workings
• Estimating Software Maintenance
Early Design and Post-Arch Models (revisited)

- **Estimated Effort:**

\[ PM = A \times (\text{Size})^E \times \prod_{i=1}^{n} EM_i \]

where \[ E = B + 0.01 \times \sum_{j=1}^{5} SF_j \]

- \[ A = 2.94 \quad B = 0.91 \]
- **Size** (discussed earlier)
- **EM:** Effort Multipliers (7 for ED, 17 for PA)
- **SF:** Scale Factors (5 for both models)
Project Scale Factors

\[ E = B + 0.01 \times \sum_{j=1}^{5} SF_j \]

- \( E \) ranges from 0.91 to 1.23
- Apply to whole project
  - Precedednedness
  - Development flexibility
  - Architecture/ risk resolution
  - Team cohesion
  - Process maturity (derived from SEI SW-CMM)
## Project Scale Factors (cont.)

<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>largely unprecedented</td>
<td>somewhat unprecedented</td>
<td>generally familiar</td>
<td>largely familiar</td>
<td>thoroughly familiar</td>
</tr>
<tr>
<td>SF₁:</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>FLEX</td>
<td>rigorous</td>
<td>occasional relaxation</td>
<td>some relaxation</td>
<td>general conformity</td>
<td>some conformity</td>
<td>general goals</td>
</tr>
<tr>
<td>SF₂:</td>
<td>5.07</td>
<td>4.05</td>
<td>3.04</td>
<td>2.03</td>
<td>1.01</td>
<td>0.00</td>
</tr>
<tr>
<td>RESL</td>
<td>little (20%)</td>
<td>some (40%)</td>
<td>often (60%)</td>
<td>generally (75%)</td>
<td>mostly (90%)</td>
<td>full (100%)</td>
</tr>
<tr>
<td>SF₃:</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</td>
<td>very difficult interactions</td>
<td>some difficult cooperative interactions</td>
<td>basically cooperative interactions</td>
<td>largely cooperative</td>
<td>highly cooperative</td>
<td>seamless interactions</td>
</tr>
<tr>
<td>SF₄:</td>
<td>5.48</td>
<td>4.38</td>
<td>3.29</td>
<td>2.19</td>
<td>1.10</td>
<td>0.00</td>
</tr>
<tr>
<td>PMAT</td>
<td>SW-CMM Level 1 Lower</td>
<td>SW-CMM Level 1 Upper</td>
<td>SW-CMM Level 2</td>
<td>SW-CMM Level 3</td>
<td>SW-CMM Level 4</td>
<td>SW-CMM Level 5</td>
</tr>
<tr>
<td>SF₅:</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>
<tr>
<td>EPML</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

or the Equivalent Process Maturity Level
Equivalent Process Maturity Level

**Requirements Management KPA**: involves establishing and maintaining an agreement with the customer on the requirements for the software project.

- **Goal 1**: System requirements allocated to software are controlled to establish a baseline for software engineering and management use.
  - Almost Always (over 90% of the time)
  - Frequently (about 60% to 90% of the time)
  - About Half (about 40% to 60% of the time)
  - Occasionally (about 10% to 40% of the time)
  - Rarely if ever (less than 10% of the time)
  - Does not apply
  - Do not know

- **Goal 2**: Software plans, products, and activities are kept consistent with the system requirements allocated to software.

After each KPA is rated, the rating level is weighted:
- 100% for Almost Always
- 75% for Frequently
- 50% for About Half
- 25% for Occasionally
- 1% for Rarely If Ever.

\[
EPML = 5 \times \left( \frac{\sum_{i=1}^{n} \text{KPA}\%_i}{100} \right) \cdot \frac{1}{n}
\]
Cost Drivers

• Cost drivers produce the $EM_i$ values in the COCOMO II effort equation

• Post-Architecture Model has 17 cost drivers
  – Product
  – Platform
  – Personnel
  – Project

• Early Design Model has 7 cost drivers

• All cost drivers except Required Development Schedule can be applied to subsystems or modules
## Post-Arch Cost Drivers: Product

<table>
<thead>
<tr>
<th>Cost Drivers</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>Required Reliability (RELY)</td>
<td>slight inconvenience</td>
<td>low, easily recoverable losses</td>
<td>moderate, easily recoverable losses</td>
<td>high financial loss</td>
<td>risk to human life</td>
<td></td>
</tr>
<tr>
<td>Database Size (DATA)</td>
<td>DB bytes / Pgm SLOC &lt; 10</td>
<td>10 ≤ D/P &lt; 100</td>
<td>100 ≤ D/P &lt; 1000</td>
<td>D/P &gt; 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Complexity (CPLX)</td>
<td>see Complexity Table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developed for Reusability (RUSE)</td>
<td>none</td>
<td>across project</td>
<td>across program</td>
<td>across product line</td>
<td>across multiple product lines</td>
<td></td>
</tr>
<tr>
<td>Documentation Match to LC Needs (DOCU)</td>
<td>Many life-cycle needs uncovered</td>
<td>Some life-cycle needs uncovered</td>
<td>Right-sized to life-cycle needs</td>
<td>Excessive for life-cycle needs</td>
<td>Very excessive for life-cycle needs</td>
<td></td>
</tr>
</tbody>
</table>
# Post-Arch Cost Driver: Complexity

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Low</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>High</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Very High</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Extra High</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Post-Arch Cost Drivers: Platform

<table>
<thead>
<tr>
<th>Cost Drivers</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>Execution Time Constraint (TIME)</td>
<td></td>
<td>≤ 50% use of available execution time</td>
<td>70%</td>
<td>85%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>Main Storage Constraint (STOR)</td>
<td></td>
<td>≤ 50% use of available storage</td>
<td>70%</td>
<td>85%</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>Platform Volatility (PVOL)</td>
<td>major change every 12 mo.; minor change every 1 mo.</td>
<td>major: 6 mo.; minor: 2 wk.</td>
<td>major: 2 mo.; minor: 1 wk.</td>
<td>major: 2 wk.; minor: 2 days</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Post-Arch Cost Drivers: Personnel

<table>
<thead>
<tr>
<th>Cost Drivers</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 Capability (ACAP)</td>
<td>15th percentile</td>
<td>35th percentile</td>
<td>55th percentile</td>
<td>75th percentile</td>
<td>90th percentile</td>
<td>n/a</td>
</tr>
<tr>
<td>Programmer Capability (PCAP)</td>
<td>15th percentile</td>
<td>35th percentile</td>
<td>55th percentile</td>
<td>75th percentile</td>
<td>90th percentile</td>
<td>n/a</td>
</tr>
<tr>
<td>Personnel Continuity (PCON)</td>
<td>48% / year</td>
<td>24% / year</td>
<td>12% / year</td>
<td>6% / year</td>
<td>3% / year</td>
<td>n/a</td>
</tr>
<tr>
<td>Application Experience (APEX)</td>
<td>≤ 2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>3 years</td>
<td>6 years</td>
<td>n/a</td>
</tr>
<tr>
<td>Platform Experience (PLEX)</td>
<td>≤ 2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>3 years</td>
<td>6 year</td>
<td>n/a</td>
</tr>
<tr>
<td>Language and Tool Experience (LTEX)</td>
<td>≤ 2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>3 years</td>
<td>6 year</td>
<td>n/a</td>
</tr>
</tbody>
</table>
# Post-Arch Cost Drivers: Project

<table>
<thead>
<tr>
<th>Cost Drivers</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><strong>Use of Software Tools (TOOL)</strong></td>
<td>edit, code, debug</td>
<td>simple, frontend, backend</td>
<td>basic lifecycle tools, moderately integrated</td>
<td>strong, mature lifecycle tools, moderately integrated</td>
<td>strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse</td>
<td></td>
</tr>
<tr>
<td><strong>Multisite Development</strong></td>
<td>International</td>
<td>Multi-city and multi-company</td>
<td>Multi-city or multi-company</td>
<td>Same city or metro area</td>
<td>Same building or complex</td>
<td>Fully collocated</td>
</tr>
<tr>
<td><strong>Collocation Communication (SITE)</strong></td>
<td>Some phone, mail</td>
<td>Individual phone, FAX</td>
<td>Narrow-band email</td>
<td>Wide-band electronic communication</td>
<td>Wide-band elect. comm, occasional video conf.</td>
<td>Interactive multimedia</td>
</tr>
<tr>
<td><strong>Required Development Schedule</strong></td>
<td>75% of nominal</td>
<td>85% of nominal</td>
<td>100% of nominal</td>
<td>130% of nominal</td>
<td>160% of nominal</td>
<td></td>
</tr>
</tbody>
</table>

## Notes
- **Use of Software Tools (TOOL)**
  - Edit, code, debug
  - Simple, frontend, backend CASE, little integration
  - Basic lifecycle tools, moderately integrated
  - Strong, mature lifecycle tools, moderately integrated
  - Strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse

- **Multisite Development Collocation Communication (SITE)**
  - International
  - Some phone, mail
  - Multi-city and multi-company
  - Multi-city or multi-company
  - Same city or metro area
  - Wide-band electronic communication
  - Wide-band elect. comm, occasional video conf.
  - Fully collocated

- **Required Development Schedule (SCED)**
  - 75% of nominal
  - 85% of nominal
  - 100% of nominal
  - 130% of nominal
  - 160% of nominal
Cost Driver: Required Development Schedule

- Required Development Schedule is the only cost driver that is applied project wide.
- It appears in both the Post-Architecture and Early Design models.
- *Nominal* is the duration estimate from $TDEV_{NS}$ (as shown in the opening example).

<table>
<thead>
<tr>
<th>Cost Driver</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>Required Development Schedule (SCED)</td>
<td>75% of nominal</td>
<td>85% of nominal</td>
<td>100% of nominal</td>
<td>130% of nominal</td>
<td>160% of nominal</td>
<td>n/a</td>
</tr>
</tbody>
</table>


## COCOMO II.2000 Post-Architecture Calibrated values

Baseline Effort Constants: \( A = 2.94; \quad B = 0.91 \)
Baseline Schedule Constants: \( C = 3.67; \quad D = 0.28 \)

<table>
<thead>
<tr>
<th>Cost Driver</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>RELY</td>
<td>0.82</td>
<td>0.92</td>
<td>1.00</td>
<td>1.10</td>
<td>1.26</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>0.90</td>
<td>1.00</td>
<td>1.14</td>
<td>1.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPLX</td>
<td>0.73</td>
<td>0.87</td>
<td>1.00</td>
<td>1.17</td>
<td>1.34</td>
<td>1.74</td>
</tr>
<tr>
<td>RUSE</td>
<td>0.95</td>
<td>1.00</td>
<td>1.07</td>
<td>1.15</td>
<td>1.24</td>
<td></td>
</tr>
<tr>
<td>DOCU</td>
<td>0.81</td>
<td>0.91</td>
<td>1.00</td>
<td>1.11</td>
<td>1.23</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>1.00</td>
<td>1.11</td>
<td>1.29</td>
<td>1.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOR</td>
<td>1.00</td>
<td>1.05</td>
<td>1.17</td>
<td>1.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PVOL</td>
<td>0.87</td>
<td>1.00</td>
<td>1.15</td>
<td>1.30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## COCOMO II.2000 Post-Architecture Calibrated values

<table>
<thead>
<tr>
<th>Cost Driver</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>ACAP</td>
<td>1.42</td>
<td>1.19</td>
<td>1.00</td>
<td>0.85</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>PCAP</td>
<td>1.34</td>
<td>1.15</td>
<td>1.00</td>
<td>0.88</td>
<td>0.76</td>
<td></td>
</tr>
<tr>
<td>PCON</td>
<td>1.29</td>
<td>1.12</td>
<td>1.00</td>
<td>0.90</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>APEX</td>
<td>1.22</td>
<td>1.10</td>
<td>1.00</td>
<td>0.88</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>PLEX</td>
<td>1.19</td>
<td>1.09</td>
<td>1.00</td>
<td>0.91</td>
<td>0.85</td>
<td></td>
</tr>
<tr>
<td>LTEX</td>
<td>1.20</td>
<td>1.09</td>
<td>1.00</td>
<td>0.91</td>
<td>0.84</td>
<td></td>
</tr>
<tr>
<td>TOOL</td>
<td>1.17</td>
<td>1.09</td>
<td>1.00</td>
<td>0.90</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>SITE</td>
<td>1.22</td>
<td>1.09</td>
<td>1.00</td>
<td>0.93</td>
<td>0.86</td>
<td>0.80</td>
</tr>
<tr>
<td>SCED</td>
<td>1.43</td>
<td>1.14</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>
# Early Design versus Post-Arch Cost Drivers

<table>
<thead>
<tr>
<th>Early Design Cost Drivers</th>
<th>Counterpart Combined Post-Architecture Cost Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Reliability and Complexity (RCPX)</td>
<td>RELY, DATA, CPLX, DOCU</td>
</tr>
<tr>
<td>Developed for Reusability (RUSE)</td>
<td>RUSE</td>
</tr>
<tr>
<td>Platform Difficulty (PDIF)</td>
<td>TIME, STOR, PVOL</td>
</tr>
<tr>
<td>Personnel Capability (PERS)</td>
<td>ACAP, PCAP, PCON</td>
</tr>
<tr>
<td>Personnel Experience (PREX)</td>
<td>APEX, PLEX, LTEX</td>
</tr>
<tr>
<td>Facilities (FCIL)</td>
<td>TOOL, SITE</td>
</tr>
<tr>
<td>Required Development Schedule (SCED)</td>
<td>SCED</td>
</tr>
</tbody>
</table>
**Effort Distribution**

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Life Cycle Objectives (LCO)</th>
<th>Life Cycle Architecture (LCA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Requirements</td>
<td>Product Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Detailed Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code &amp; Unit Test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integration &amp; Test</td>
</tr>
</tbody>
</table>

- **Small project with low exponent**
- **Large project with high exponent**

<table>
<thead>
<tr>
<th>E=1.05</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSLOC</td>
<td>6 6 6 6 6</td>
<td>16 16 16 16 16</td>
<td>26 25 24 23</td>
<td>42 40 38 36</td>
</tr>
<tr>
<td>MBASE</td>
<td>6 (2 – 15)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E=1.12</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSLOC</td>
<td>7 7 7 7 7</td>
<td>17 17 17 17 17</td>
<td>27 26 25 24 23</td>
<td>37 35 33 31 29</td>
</tr>
<tr>
<td>MBASE</td>
<td>7 (2 – 15)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E=1.20</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
<th>2 8 32 128 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSLOC</td>
<td>8 8 8 8 8</td>
<td>18 18 18 18 18</td>
<td>28 27 26 25 24</td>
<td>32 30 28 26 24</td>
</tr>
<tr>
<td>MBASE</td>
<td>8 (2 – 15)</td>
<td>24 (20 – 28)</td>
<td>76 (72 – 80)</td>
<td></td>
</tr>
</tbody>
</table>

10/27/00

Copyright 2000 USC-CSE
Post-Arch Model Example (cont)

- \( PM_{NS} = 2.94(100)(0.91+0.01*24) = 586.61 \) person months
  - \( E = 1.15 \)

- Effort Distribution:
  - Requirements: 41 PM
  - Product Design: 100 PM
  - Detailed Design: 141 PM
  - Code & Unit Test: 182 PM
  - Integration & Test: 164 PM
COCOMO II Model Overview

• COCOMO II Overview
• Sizing the Application
• Estimating Effort
• Estimating Schedule
• Understanding model workings
• Estimating Software Maintenance
Early Design and Post-Arch Models (revisited)

• Estimated Duration (schedule):

\[ TDEV = \left[ C \times \left( PM_{NS} \right)^D + 0.2 \times 0.1 \times \sum_{j=1}^{5} SF_j \right] \times \frac{SCED\%}{100} \]

- TDEV: Time to Develop
- C = 3.67 \quad D = 0.28
- Where PM_{NS} is the estimated nominal-schedule person months (excludes SCED cost driver effects)
- SF: Scale Factors (5 for both models)
- SCED\%: Percentage of schedule compression from nominal (from the SCED cost driver)
Schedule Distribution

Distribution of Schedule by Size and Scale Factor

- E=1.05
- E=1.12
- E=1.20
- MBASE

Large project with high exponent
Small project with low exponent
Post-Arch Model Example (cont)

- $TDEV_{NS} = 3.67(586.6)^{(0.28+0.2*0.01*24)} = 29.7$ months

- Schedule Distribution:
  - Requirements: 5 Months
  - Programming: 8 Months
  - Product Design: 6 Months
  - Integration & Test: 6 Months
COCOMO II Model Overview

• COCOMO II Overview
• Sizing the Application
• Estimating Effort
• Estimating Schedule
• Understanding model workings
• Estimating Software Maintenance
Model Breadth: Life Cycle Phases

COCOMO II Estimation Endpoints

MBASE/RUP
Inception
Elaboration
Construction
Transition

Waterfall
Plans and Requirements
Preliminary (Product) Design
Detailed Design
Code and Unit Test
Integration and Test

Most likely model to use:
Early Design Model
Post-Architecture Model

Most likely model to use:

Early Design Model
Post-Architecture Model
# MBASE and RUP Phase Distribution

<table>
<thead>
<tr>
<th>Phase (Endpoints)</th>
<th>MBASE</th>
<th>RUP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Effort%</td>
<td>Schedule%</td>
</tr>
<tr>
<td><strong>Inception (IRR to LCO)</strong></td>
<td>6 (2-15)</td>
<td>12.5 (2-30)</td>
</tr>
<tr>
<td><strong>Elaboration (LCO to LCA)</strong></td>
<td>24 (20-28)</td>
<td>37.5 (33-42)</td>
</tr>
<tr>
<td><strong>Construction (LCA to IOC)</strong></td>
<td>76 (72-80)</td>
<td>62.5 (58-67)</td>
</tr>
<tr>
<td><strong>Transition (IOC to PRR)</strong></td>
<td>12 (0-20)</td>
<td>12.5 (0-20)</td>
</tr>
<tr>
<td><strong>Totals:</strong></td>
<td>118</td>
<td>125</td>
</tr>
</tbody>
</table>
Model Depth: Activity WBS

Subsystem Activities

Management
- Cost, Schedule, Performance Management
- Contract Management
- Subcontract Management
- Customer Interface
- Branch Office Management
- Management Reviews & Audits

System Engineering
- Software Requirements
- Product Design
- Configuration Management
- End Item Acceptance
- Quality Assurance

Programming
- Detailed Design
- Code and Unit Test
- Integration

Test & Evaluation
- Product Test
- Acceptance Test
- Test Support

Data
- Manuals
Impact of Application Experience

The level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team’s equivalent level of experience with this type of application.

<table>
<thead>
<tr>
<th>Descriptors:</th>
<th>2 ≤ months</th>
<th>6 months</th>
<th>1 year</th>
<th>3 years</th>
<th>6 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating Levels</td>
<td>Very Low</td>
<td>Low</td>
<td>Nominal</td>
<td>High</td>
<td>Very High</td>
</tr>
<tr>
<td>Effort Multipliers</td>
<td>1.22</td>
<td>1.10</td>
<td>1.00</td>
<td>0.88</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Effect on Effort: +11% | +10% | -13% | -9%
Effects of Schedule Compression

- 15% Schedule Compression:
  - requires 14% additional staff
  - effort multiplier increase 1.14
- 25% Schedule Compression:
  - requires 43% additional staff
  - effort multiplier increase 1.43

Maximum schedule compression is 25%
Extending the schedule does not change the staffing requirement
Model Particulars

• Size is for delivered product size
• A Person Month is 152 hours of effort
• Calibration is from the end of requirements analysis to the end of integration and test. Other phases are “add-ons”
• The model is more accurate if it is calibrated to local development conditions
COCOMO II Model Overview

• COCOMO II Overview
• Sizing the Application
• Estimating Effort
• Estimating Schedule
• Understanding model workings
• Estimating Software Maintenance
Sizing Software Maintenance -1

- Apply Scale Factors, E, to the size of code being changed (rather than the whole product being modified)
- Account for effects of modification to an existing body of code with software understanding (SU) and programmer unfamiliarity (UNFM)
- Excludes
  - Major product rebuilds (over 50%)
  - Development of sizeable interfaces (over 20%)
Sizing Software Maintenance -2

\[ Size_M = \left[ \left( \frac{\text{Base Code Size}}{\text{Base Code Size}} \right) \times \text{MCF} \right] \times \text{MAF} \]

where \( \text{MCF} = \left( \frac{\text{Size Added}}{\text{Base Code Size}} + \frac{\text{Size Modified}}{\text{Base Code Size}} \right) \times \text{MAF} \)

where \( \text{MAF} = 1 + \left( 100 \times \frac{\text{SU}}{\text{UNFM}} \right) \)

- **MCF**: Maintenance Change Factor is the percentage of change to the base code
- **MAF**: Maintenance Adjustment Factor is used to adjust the effective maintenance size to account for software understanding and programmer familiarity with the software being maintained
Maintenance Considerations

• SCED cost driver is not used - maintenance cycle is considered fixed
• RUSE cost driver is not used - maintaining a component is balanced by is careful design, documentation and testing
• RELY cost driver has a different set of effort multipliers (EM). Depends on the required reliability under which the product was developed
Maintenance Model

- Maintenance effort and schedule estimation:

\[ PM_M = A \times (\text{Size}_M)^E \times \prod_{i=1}^{15} EM_i \]

\[ FSPM = \frac{PM_M}{TM} \]

- \( A = 2.94 \), \( E = \) scaling exponent
- \( \text{Size}_M = \) size of code being changed
- \( TM = \) Fixed time for maintenance
- \( FSPM = \) Full Time Software Personnel for Maintenance
Questions?

Software Cost Estimation with COCOMO II

Barry W. Boehm
Chris Abts
A. Winsor Brown
Sunita Chulani
Bradford K. Clark
Ellis Horowitz
Ray Madachy
Donald Reifer
Bert Steece

Printice Hall PTR
Upper Saddle River, NJ