COCOMO II
Exercise Solution

EEE492A 2007
References: [HvV §7.4]

Maj Ron Smith
Royal Military College of Canada
Electrical and Computer Engineering
smith-r@rmc.ca tarpit.rmc.ca/smithr/

COCOMO II -Exercise Scenario

• use COCOMO-II model to estimate the effort and schedule for the Personal Income Tax (PIT) software application
  • approximately 648 unadjusted function points
• given that requirements prototyping and a basic architectural design have been completed, use the Post-Architectural model
  \[ PM_{NS} = a \times \text{Size}^b \times \prod_{i=1}^{16} EM_i \]  
  \( (i = 1 \text{ to } 16) \)
• certain scale and effort multipliers can be attained from the scenario description, others must be assumed to be nominal
Scaling Factors

Table 2.48  Scale Factors for COCOMO II Models

<table>
<thead>
<tr>
<th>Scale 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>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>FLEX</td>
<td>rigorous</td>
<td>occasional relaxation</td>
<td>some relaxation</td>
<td>general conformity</td>
<td>generally familiar</td>
<td>general conformity</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>TEAM</td>
<td>very difficult interactions</td>
<td>some difficult interactions</td>
<td>basically cooperative interactions</td>
<td>largely cooperative</td>
<td>highly cooperative</td>
<td>seamless interactions</td>
</tr>
<tr>
<td>PMAT</td>
<td>SW-CMM Level 1 Lower</td>
<td>SW-CMM Level 1 Lower</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>
</tbody>
</table>

or the estimated Equivalent Process Maturity Level (EFML)

Scaling Factors

- cause an exponential cost increase

<table>
<thead>
<tr>
<th>Cost Drivers</th>
<th>Very Low</th>
<th>Low</th>
<th>Nom</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedentedness (PREC)</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Development flexibility (FLEX)</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Architecture/risk resolution (RESL)</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Team cohesion (TEAM)</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Process Maturity (PMAT)</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## Effort Multipliers (Post-Architecture)

### Table 2.49 Cost Driver Ratings for Post-Architecture Model

<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>RELY</td>
<td>slight</td>
<td>low, easily</td>
<td>moderate, easily</td>
<td>high financial</td>
<td>risk to human life</td>
<td></td>
</tr>
<tr>
<td></td>
<td>inconvenience</td>
<td>recoverable losses</td>
<td>losses</td>
<td>loss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>Testing 0.06</td>
<td>bytes/Function</td>
<td>$10 \leq D/P &lt; 100$</td>
<td>$100 \leq D/P &lt; 1000$</td>
<td>$D/P &gt; 1000$</td>
<td></td>
</tr>
<tr>
<td>CPLX</td>
<td>see Table 2.19</td>
<td>no new &amp; SLOC &lt; 10</td>
<td>across project</td>
<td>across program</td>
<td>across product line</td>
<td></td>
</tr>
<tr>
<td>RUSE</td>
<td>across multiple projects</td>
<td>across multiple products</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCU</td>
<td>Many lifecycle needs</td>
<td>uncovered</td>
<td>Right-sized to</td>
<td>Excessive for</td>
<td>Very excessive for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lifecycle needs</td>
<td>lifecycle needs</td>
<td>lifecycle needs</td>
<td>lifecycle needs</td>
<td>lifecycle needs</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>$\leq 50%$ use of available execution time</td>
<td>$70%$</td>
<td>$85%$</td>
<td>$95%$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOR</td>
<td>$\leq 50%$ use of available storage</td>
<td>$70%$</td>
<td>$85%$</td>
<td>$95%$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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>

## Effort Multipliers (Post-Architecture)

### Table 2.49 Cost Driver Ratings for Post-Architecture Model

<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>ACAP</td>
<td>50th percentile</td>
<td>30th percentile</td>
<td>55th percentile</td>
<td>75th percentile</td>
<td>90th percentile</td>
<td></td>
</tr>
<tr>
<td>PCAP</td>
<td>50th percentile</td>
<td>24th percentile</td>
<td>40th percentile</td>
<td>60th percentile</td>
<td>90th percentile</td>
<td></td>
</tr>
<tr>
<td>PCON</td>
<td>48%/year</td>
<td>24%/year</td>
<td>42%/year</td>
<td>60%/year</td>
<td>90%/year</td>
<td></td>
</tr>
<tr>
<td>APEX</td>
<td>2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>2 years</td>
<td>6 years</td>
<td></td>
</tr>
<tr>
<td>PLEX</td>
<td>2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>3 years</td>
<td>6 years</td>
<td></td>
</tr>
<tr>
<td>LESP</td>
<td>1 year</td>
<td>1 year</td>
<td>2 years</td>
<td>3 years</td>
<td>6 years</td>
<td></td>
</tr>
<tr>
<td>TOOL</td>
<td>simple, front-end, back-end</td>
<td>CASE, little integration</td>
<td>basic life cycle tools, moderately integrated</td>
<td>strong, mature life cycle tools, fully integrated</td>
<td>strong, mature, proactive life cycle tools, fully integrated</td>
<td></td>
</tr>
<tr>
<td>SITE Cordination</td>
<td>International</td>
<td>Multi-city or multi-company</td>
<td>Multi-city or multi-company</td>
<td>Same city or metro area</td>
<td>Same city or metro area</td>
<td></td>
</tr>
<tr>
<td>SITE Communi-cation</td>
<td>Some phone, email</td>
<td>Narrow-band communication</td>
<td>Wide-band electronic communication</td>
<td>Wide-band electronic communication, occasional video conference</td>
<td>Wide-band electronic communication, occasional video conference</td>
<td></td>
</tr>
<tr>
<td>SCED</td>
<td>75% of nominal</td>
<td>88% of nominal</td>
<td>100% of nominal</td>
<td>130% of nominal</td>
<td>160% of nominal</td>
<td>Fully collocated (interactive multimedia)</td>
</tr>
</tbody>
</table>
### Effort Multipliers (Post-Architecture)

- **Personnel factors**

<table>
<thead>
<tr>
<th>Cost Drivers</th>
<th>Very Low</th>
<th>Low</th>
<th>Nom</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyst capability (ACAP)</td>
<td>1.50</td>
<td>1.22</td>
<td>1.00</td>
<td>0.83</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>Programmer .. (PCAP)</td>
<td>1.37</td>
<td>1.16</td>
<td>1.00</td>
<td>0.87</td>
<td>0.74</td>
<td></td>
</tr>
<tr>
<td>Application experience (APEX)</td>
<td>1.22</td>
<td>1.10</td>
<td>1.00</td>
<td>0.89</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>Platform .. (PLEX)</td>
<td>1.24</td>
<td>1.10</td>
<td>1.00</td>
<td>0.92</td>
<td>0.84</td>
<td></td>
</tr>
<tr>
<td>Language/tool .. (LTEX)</td>
<td>1.25</td>
<td>1.12</td>
<td>1.00</td>
<td>0.88</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>Personnel continuity (PCON)</td>
<td>1.24</td>
<td>1.10</td>
<td>1.00</td>
<td>0.92</td>
<td>0.84</td>
<td></td>
</tr>
</tbody>
</table>

- **Project factors**

<table>
<thead>
<tr>
<th>Cost Drivers</th>
<th>Very Low</th>
<th>Low</th>
<th>Nom</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of software tools (TOOL)</td>
<td>1.24</td>
<td>1.12</td>
<td>1.00</td>
<td>0.86</td>
<td>0.72</td>
<td></td>
</tr>
<tr>
<td>Multi-site development (SITE)</td>
<td>1.25</td>
<td>1.10</td>
<td>1.00</td>
<td>0.92</td>
<td>0.84</td>
<td>0.78</td>
</tr>
<tr>
<td>Required development schedule (SCED)</td>
<td>1.29</td>
<td>1.10</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>
Calculating Estimated Effort

\[ \text{PM}_{NS} = a \times \text{Size}^b \times \prod \text{EM}_i \quad (i = 1 \text{ to } 16) \]

where

- \( a = 2.94 \) (calibrated from 161 projects)
- \( b = 1.01 + 0.01 \times \sum \text{SF}_j \quad (j = 1 \text{ to } 5) \)
  \[ = 1.01 + 0.01 \times (1+1+3+4+3) = 1.13 \]
- \( \prod \text{EM}_i = \text{ACAP} \times \text{APEX} \times \text{PCON} \times \text{SITE} \)
  \[ = 1.22 \times 0.89 \times 1.17 \times 0.84 = 1.067 \]

Size = 648 UFPs * 23 Java SLOC/UFP
\[ = 14.9 \text{ kSLOC} \]

therefore

\[ \text{PM}_{NS} = 2.94 \times 14.9^{(1.13)} \times 1.067 \]
\[ = 66.4 \text{ person-months} \]

COCOMO II Exercise - Conclusion

- recall that the total effort for PIT was 66 person-months (based upon a nominal schedule)
- now determine the nominal development time

\[ TDEV = [c \times (\text{PM}_{NS})^d] \]

where

- \( c = 3.67 \)
- \( d = 0.28 + 0.2 \times [b - 1.01] \)

and recall

- \( b = 1.01 + 0.01(1+1+3+4+3) = 1.13 \)

therefore

\[ TDEV = 3.67 \times (66.4)^{0.304} \]
\[ = 13.1 \text{ calendar months} \]
**Sensitivity Analysis (1)**

- compare the effect that documentation requirements might have upon the project effort

- for minimal doc requirements (DOCU = 0.89)
  \[
  \Pi EM_i = ACAP \times APEX \times PCON \times SITE \times DOCU \\
  = 1.067 \times 0.89 = 0.950
  \]
  or more simply,
  \[
  PM_{NS} = 66.4 \times 0.89 = 59 \text{ person-months}
  \]

- and for excessive doc requirements (DOCU = 1.13)
  \[
  PM_{NS} = 66.4 \times 1.13 = 75 \text{ person-months}
  \]

**Sensitivity Analysis (2)**

- what savings might be achieved if you were to change contracts and bring in a Level IV company? what else must you consider?

  now \[ b = 1.01 + 0.01 \times (1+1+3+4+1) = 1.11 \]

  and

  \[
  PM_{NS} = 2.94 \times 14.9^{(1.11)} \times 1.067 \\
  = 62.9 \text{ person-months}
  \]

- a possible savings of 5%, but what about other factors such as application experience, ...
exercise continued (1)

- but your boss wants it in 6 months!
  - no problem just add people, right?

- apply maximum schedule compression
  - will you now satisfy yours bosses demand?  
  - how much more (roughly) will this cost?

assuming max schedule compression (75% of nominal)

\[ TDEV = [c \times (PM_{NS})^d] \times \text{Schedule Compression} \]
\[ = 13.1 \times (.75) \]
\[ = 9.8 \text{ calendar months} \]

- you are part way there! but at what cost?

exercise continued (2)

- go back to the original effort equation, but this time take into account the fact that it is not a nominal schedule

\[ PM_{NS} = 2.94 \times 14.9^{(1.13)} \times \prod EM_i \]

but now \[ \prod EM_i = ACAP \times APEX \times PCON \times SITE \times SCED \]
\[ = 1.067 \times (1.29) = 1.376 \]

therefore \[ PM = 2.94 \times 14.9^{(1.13)} \times 1.376 \]
\[ = 85.6 \text{ person-months} \]

- a cost increase in total effort of almost 30%
exercise continued (3)

• if we investigate relaxing the schedule to the maximum of 160% of nominal, we get

\[
TDEV = [c \times (PM_{NS})^d] \times SCED\%/100
\]

\[
= 13.1 \times 1.60
\]

\[
= 21.0 \text{ calendar months}
\]

\[
PM_{NS} = 2.94 \times 14.9^{(1.13)} \times \Pi EM_i
\]

with \[
\Pi EM_i = ACAP \times APEX \times PCON \times SITE \times SCED
\]

\[
= 1.067 \times 1.00 = 1.067
\]

finally \[
PM = 66.4 \text{ person-months}
\]

• relaxing the schedule did not save any effort??

Reuse Analysis

• the question is: will the effort involved in reusing the CTRC be sufficiently less than the effort to develop the same functionality from scratch during the PIT development?

• the CTRC can satisfy 20% of the required functionality for PIT (or about 130 UFPs)
  • the CTRC consists of 4000 source lines of Java
  • 130 UFPs represents approximately 3.0 kSLOC (Java)

• using the COCOMO II Reuse model, determine the reuse size estimate

• compare this reuse size to the expected development size of code
  • is \( Size_{RU} < Size_D \)?
Reuse Software Guidelines

Determining Reuse Size

\[ \text{Size}_{\text{RU}} = \frac{\text{Size}_{\text{A}} \times [\text{AA} + \text{AAF} + (\text{SU} \times \text{UNFM})]}{100} \]

where

\[ \text{Size}_{\text{A}} = 4.0 \text{ kSLOC} \]

and

AE - % of assessment and assimilation
AAF - adaptation adjustment factor (%)
SU - % software understanding increment (10 - 50)
UNFM - programmer unfamiliarity (0.0 - 1.0)
Adaptation Adjustment Factor

- recall, default efforts for each phase are assumed to be 40%, 30% and 30% respectively

\[
\text{AAF} = 0.4 \text{ DM} + 0.3 \text{ CM} + 0.3 \text{ IM} \\
= 0.4(20) + 0.3(30) + 0.3(25) \\
= 24.5\%
\]

note that,

IM - is the incremental % of effort required to integrate the adapted software into the product as compared to integrating a newly developed equivalent sized piece of software

Assessment and Assimilation Increment

<table>
<thead>
<tr>
<th>AA Increment</th>
<th>Level of AA Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>2</td>
<td>Basic module search and documentation</td>
</tr>
<tr>
<td>4</td>
<td>Some module Test and Evaluation (T&amp;E), documentation</td>
</tr>
<tr>
<td>6</td>
<td>Considerable module T&amp;E, documentation</td>
</tr>
<tr>
<td>8</td>
<td>Extensive module T&amp;E, documentation</td>
</tr>
</tbody>
</table>

in percentage (%)
Software Understanding Increment

Table 2.5  Rating Scale for Software Understanding Increment (SU)

<table>
<thead>
<tr>
<th>SU Increment to ISLOC</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Programmer Unfamiliarity Factor

Table 2.7  Rating Scale for Programmer Unfamiliarity (UNFM)

<table>
<thead>
<tr>
<th>UNFM Increment</th>
<th>Level of Unfamiliarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Completely familiar</td>
</tr>
<tr>
<td>0.2</td>
<td>Mostly familiar</td>
</tr>
<tr>
<td>0.4</td>
<td>Somewhat familiar</td>
</tr>
<tr>
<td>0.6</td>
<td>Considerably familiar</td>
</tr>
<tr>
<td>0.8</td>
<td>Mostly unfamiliar</td>
</tr>
<tr>
<td>1.0</td>
<td>Completely unfamiliar</td>
</tr>
</tbody>
</table>
Is Reuse Warranted?

\[
\text{Size}_{RU} = \frac{\text{Size}_A [\text{AA} + \text{AAF} + (\text{SU} \times \text{UNFM})]}{100} \\
= 4.0 \left\{ 4 + 24.5 + 20(0.8) \right\}/100 \\
= 1.78 \text{kSLOC}
\]

- therefore the reuse savings is 1.78 kSLOC versus 3.0 kSLOC or 40%
  - accordingly reuse does appear to provide meaningful savings

- other issues to consider?
  - how accurate are the redesign and recode estimates given the unfamiliarity of the programmers with CTRC

Supplemental References