Estimation of Expedited Systems Engineering Schedules

Barry Boehm, Dan Ingold, JoAnn Lane, USC-CSSE

SCEA-ISPA Joint Conference
June 2012
Outline

• Baseline: CORADMO Expedited Software Development Model
  – RAD: Rapid Application Development
  – Expedited Schedule Drivers
  – Relation to RAD Opportunity Tree

• Nominal Systems Engineering effort and schedule obtained from COSYSMO effort estimation model, cube-root effort-schedule relationship

• RAD Opportunity Tree elements reorganized around product-process-project-people-risk factors
COCOMO II RAD Extension (CORADMO)

COCOMO II cost drivers (except SCED)

Language Level, experience,...

COCOMO II

Phase Distributions (COPSEMO)

Baseline effort, schedule
Effort, schedule by stage

RAD Extension

RAD effort, schedule by phase

RVHL
DPRS
CLAB
RESL
PPOS
RCAP

Presented at the 2012 SCEA/ISPA Joint Annual Conference and Training Workshop - www.iceaaonline.com
• RAD a critical competitive strategy  
  – Market window; pace of change

• Non-RAD COCOMO II overestimates RAD schedules  
  – Need opportunity-tree cost-schedule adjustment  
  – Cube root model inappropriate for small RAD projects  
    • COCOMO II: $\text{Months} = 3.7 \sqrt[3]{\text{Person-Months}}$  
    • 27 PM $\Rightarrow 3.7 \times 3 = 11.1$ Months  
    • Small Staff size $(27/11.1 = 2.4$ people$)$ reduces cost  
      – But slow with respect to competition  
    • Larger staff size $(27/5$ people$ = 5.4$ months$)$ more competitive
RAD Opportunity Tree

Eliminating Tasks

- Development process reengineering - DPRS
- Reusing assets - RVHL
- Applications generation - RVHL
- Schedule as Independent Variable Process

Reducing Time Per Task

- Tools and automation - O
- Work streamlining (80-20) - O
- Increasing parallelism - RESL

Reducing Risks of Single-Point Failures

- Reducing failures - RESL
- Reducing their effects - RESL
- Early error elimination - RESL

Reducing Backtracking

- Process anchor points - RESL
- Improving process maturity - O
- Collaboration technology - CLAB
- Minimizing task dependencies - DPRS
- Avoiding high fan-in, fan-out - DPRS
- Reducing task variance - DPRS
- Removing tasks from critical path - DPRS

Activity Network Streamlining

Increasing Effective Workweek

- 24x7 development - PPOS
- Nightly builds, testing - PPOS
- Weekend warriors - PPOS

Better People and Incentives

- RAD Capability and experience - RCAP
- O: covered by
## RCAP:RAD Capability of Personnel

<table>
<thead>
<tr>
<th>FACTOR</th>
<th>XL</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>XH</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERS-R</td>
<td>10%</td>
<td>25%</td>
<td>40%</td>
<td>55%</td>
<td>70%</td>
<td>85%</td>
<td>95%</td>
</tr>
<tr>
<td>PREX-R</td>
<td>≤2mo</td>
<td>4 mo</td>
<td>6 mo</td>
<td>1 yr</td>
<td>3 yrs</td>
<td>6 yrs</td>
<td>10 yrs</td>
</tr>
</tbody>
</table>

### I,E, C Multipliers

<table>
<thead>
<tr>
<th></th>
<th>PM</th>
<th>M</th>
<th>P=PM/M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.20</td>
<td>1.40</td>
<td>.86</td>
</tr>
<tr>
<td></td>
<td>1.13</td>
<td>1.25</td>
<td>.90</td>
</tr>
<tr>
<td></td>
<td>1.06</td>
<td>1.12</td>
<td>.95</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>.93</td>
<td>.82</td>
<td>1.13</td>
</tr>
<tr>
<td></td>
<td>.86</td>
<td>.68</td>
<td>1.26</td>
</tr>
<tr>
<td></td>
<td>.80</td>
<td>.56</td>
<td>1.43</td>
</tr>
</tbody>
</table>

PERS-R is the Early Design Capability rating, adjusted to reflect the performers’ capability to rapidly assimilate new concepts and material, and to rapidly adapt to change.

PREX-R is the Early Design Personnel Experience rating, adjusted to reflect the performers’ experience with RAD languages, tools, components, and COTS integration.
RCAP Example

RCAP = Nominal  PM = 25, M = 5, P = 5  
The square root law: 5 people for 5 months: 25 PM

RCAP = XH  PM = 20, M = 2.8, P = 7.1  
A very good team can put on 7 people and finish in 2.8 months: 20 PM

RCAP = XL  PM = 30, M = 7, P = 4.3  
Trying to do RAD with an unqualified team makes them less efficient (30 PM) and 
gets the schedule closer to the cube root law:

(but not quite:  $3 \sqrt[3]{30 \text{ person-months}} = 9.3 \text{ months} > 7 \text{ months}$)

June 2012
Effect of RCAP on Cost, Schedule
Outline

• Baseline: CORADMO Expedited Software Development Model
  – RAD: Rapid Application Development
  – Expedited Schedule Drivers
  – Relation to RAD Opportunity Tree

➤ Nominal Systems Engineering effort and schedule obtained from COSYSMO effort estimation model, cube-root effort-schedule relationship

• RAD Opportunity Tree elements reorganized around product-process-project-people-risk factors
Basic Expedited SE Model Form

- Estimate SE effort using COSYSMO

- Estimate nominal SE schedule as $1.5 \times \text{cube root (SE effort)}$
  - Software, VLSI development schedule $= 3 \times \text{cube root (SE effort)}$
  - Roughly 50% of development schedule needed for SE

- Estimate deviations from nominal schedule using multipliers for product, process, project, people, and risk acceptance factors
  - Very Low, Low factor ratings slow down schedule
  - High, Very High and Extra High factor ratings speed up schedule
  - Factor ratings generally a weighted average of several elements
COSYSMO Operational Concept

- # Requirements
- # Interfaces
- # Scenarios
- # Algorithms
+ Volatility Factor

<table>
<thead>
<tr>
<th>Size</th>
<th>Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Effort

- Application factors
  - 8 factors
- Team factors
  - 6 factors
- Schedule driver

Volatility Factor

WBS guided by ISO/IEC 15288

June 2012
## COSYSMO Cost Drivers - Application

<table>
<thead>
<tr>
<th>TOC</th>
<th>COSYSMO Application Factor Selection</th>
<th>See Embedded Comments for Descriptions and Selection Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSYSMO Application Factor</td>
<td>Identifier</td>
<td>Current Prod. Range</td>
</tr>
<tr>
<td>Requirements Understanding</td>
<td>RQMT</td>
<td>1.73</td>
</tr>
<tr>
<td>Architecture Complexity</td>
<td>ARCH</td>
<td>1.66</td>
</tr>
<tr>
<td>Level of Service (KPP) Requirements</td>
<td>LSVC</td>
<td>2.50</td>
</tr>
<tr>
<td>Migration Complexity</td>
<td>MIGR</td>
<td>1.50</td>
</tr>
<tr>
<td>No. and Diversity of Installations/Platforms</td>
<td>INST</td>
<td>1.50</td>
</tr>
<tr>
<td>No. of Recursive Levels in the Design</td>
<td>RECU</td>
<td>1.50</td>
</tr>
<tr>
<td>Documentation to Match Lifecycle Needs</td>
<td>DOCU</td>
<td>0.67</td>
</tr>
<tr>
<td>Technology Maturity</td>
<td>TMAT</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Productivity Range (PR) is the Highest Number / Lowest Number and is an indication of the "Relative Degree of Influence" of this parameter on SE effort as currently. The "Suggested" column has no immediate impact in the COSYSMO SE Costing Mode. However, for the COSYSMO SE Data Collection Mode, it serves as a means of collecting your inputs as to what you think the "Relative Degree of Influence" of this parameter should be based upon your overall experience (not specific to the past program being characterized). If you agree with the "current" number, do nothing. If you disagree, simply overwrite the current number with a new number (n>1.0) in the appropriate cell.
Product Factor Elements

• Product simplicity (of interfaces, legacy migration, -ilities)
  – Very Low: Extremely complex; Extra High: Extremely simple

• Ability to reuse product elements
  – Very Low: None; Extra High: over 90%

• Ability to defer low-impact aspects
  – Very Low: None; Extra High: over 90%

• System definition via models vs. documents
  – Very low: None; Extra High: over 90%

• Technology maturity of key capabilities
  – Very Low: >0 Level 1-2 or >1 Level 3; Extra High: All >Level 7
Process Factor Elements

- Concurrency of OpCon, Rqts., Architecture, V&V
  - Very Low: Highly sequential; Extra High: Fully concurrent

- Process streamlining
  - Very Low: Heavily Bureaucratic; Extra High: Fully streamlined

- General SE tool support (coverage, maturity, integration: CMI)
  - Very Low: Simple tools, weak CMI; Extra High: Very strong CMI
Project Factor Elements

- **Collaboration support**
  - Very Low: Globally distributed; weak communications, data sharing
  - Extra High: Largely collocated; very strong communications, data sharing

- **Single-domain models, methods, processes, tools (MMPTs)**
  - Very Low: Simple MMPTs, weak CMI; Extra High: Very strong CMI

- **Multi-domain models, methods, processes, tools (MMPTs)**
  - Very Low: Simple MMPTs, weak CMI; Extra High: Very strong CMI
People Factor Elements

• **General-SE Knowledge, Skills, and Agility (KSA)**
  — Very Low: Very weak KSA; Extra High: Extra strong KSA

• **Single-domain Knowledge, Skills, and Agility (KSA)**
  — Very Low: Very weak KSA; Extra High: Extra strong KSA

• **Multi-domain Knowledge, Skills, and Agility (KSA)**
  — Very Low: Very weak KSA; Extra High: Extra strong KSA

• **Team compatibility**
  — Very Low: Continuous strong conflict
  — Extra High: Very strong leadership, commonality of interests
• Risk Acceptance
  – Very Low: Highly risk-averse; Extra High: Highly risk-accepting
Current Model Status

• Results similar to CORADMO for software-intensive systems
  – Considered useful for planning
  – Preparing Delphi exercise for relative parameter influence ranges for systems engineering

• Need further data for hardware-intensive systems
  – Good data and driver ratings hard to find