Optimized Project Life Cycle Cost and Performance in an Era of Constrained Budgets

PSM Workshop

Tuesday

31 July 2012
Attendees

- Bob Epps
- Francie Bis
- Cari Pullen
- Cheryl Jones
- Bob Ferguson
- Bob McCann
- Bill Golaz
- JoAnn Lane
- Jack McGarry
- Joe Dean
- Bob Chartte
Goals

• Discuss project optimization
• Add maintenance to total system optimization
  – Corrective actions
  – Enhancements
  – Technical Debt
• Establish methods to enhance parametric models to incorporate M&O for total system project optimization
Workshop results

• Discussed Optimization as it relates to project management during SW project development

• Opened discussion concerning adding maintenance to consider total project optimization
  – Cost, Schedule, or risk
  – Product value and use
  – Capability and/or capacity

• Consideration for trade-offs versus constraints
Workshop Results (cont.)

• Potential considerations for Maintenance optimization
  – Defect density
  – Fixed versus variable cost for Maintenance
  – Number of change requests
  – Life Cycle Scope

• Consider the ability to incorporate Port Folio Management
Workshop Output

• Established that Port Folio Management consideration has wide range of objective considerations that are beyond current estimation methods.
• Established a limited maintenance parametric estimation aligned to the maintenance component of the WBS presented in Life Cycle Maintenance Cost Estimation Model and Drivers workshop
• Defined method to calibrate parametric model to estimate limited maintenance.
Optimized Project Life Cycle Cost and Performance in an Era of Constrained Budgets

PSM Workshop

Tuesday July 31, 2012
Workshop Discussion Items

• Workshop discussion includes the following concepts and the software measures that will be required.
• Identify objectives as a requirement in making optimal project and design decisions
• Software Measures to support effective confidence based project plans.
  – Plans are consistent with the effort/schedule/risk estimates for the project
• Incorporation of performance based project management
  – Integration of the effort/schedule/risk estimation into project control
• Established methods to integrate project trade-offs into both planning and ongoing project control
  – Design decisions
  – Project decisions from architectural trade-offs to low level decisions such as development standard applied versus additional functionality or coding standards versus shorter development cycle.
• Technical Debt
  – Define Technical Debt
  – What are effective measurement for Technical Debt in support of project planning and execution
  – Establish an objective Technical baseline of the system defining the structural quality and health index
  – System attributes and key indicators
Workshop Goals

Developing measurements which provide the information required for Optimal project decisions

- Performance based software project measurements
- Parameters that would support cost estimation models in the evaluation of total life cycle costs including development, maintenance, sustainment, and resolution of technical debt
- Software measurements to identify Technical Debt
- Identification of software measurements to support cost estimation of Technical Debt as component of total system costs.
Optimization

• The maximizing or minimizing of a given function possibly subject to some type of constraints.

• (systems engineering) Broadly, the efforts and processes of making a decision, a design, or a system as perfect, effective, or functional as possible. Narrowly, the specific methodology, techniques, and procedures used to decide on the one specific solution in a defined set of possible alternatives that will best satisfy a selected criterion. Also known as system optimization.
Software & IT Systems Are About Business Value

- Software economics should provide methods for analyzing the choices software projects must make.” Leon Levy...
- Business economics should provide methods for analyzing choices in which projects to fund
- Project economics should provide methods for project trade-offs
Project Estimation

• Total systems level estimation
• Project initiation
• Project planning
• Project execution (Project Optimization)
• Technical Debt
• Maintenance and Sustainment
An Estimate Defined

• An **estimate** is the most knowledgeable statement you can make **at a particular point in time** regarding:
  – Effort / Cost
  – Schedule
  – Staffing
  – Risk
  – Reliability

• Estimates more precise with progress

• **A WELL FORMED ESTIMATE IS A DISTRIBUTION**
Control Theory — Project Management

• **Feedback control requires continual collection of measurements reflecting performance of “the system”**

• Feedback control systems typically implement a model
  – Best possible math representation of the real world
  – Provides the basis to evaluate alternative control actions available
  – Typically control inputs are the result of analysis supported by a math model representing the “real world”

• Software Project Management control system can be implemented in the same fashion
  – The parametric models used to establish plan (effort, staffing, schedules) can provide basis for Project Management “feedback” control
  – Maintaining the models “after the estimate” allow the models to continually improve the fidelity of the information provided
Software Project Management

• Software Project Management
  – Establish objectives
  – Apply resources
  – Obtain project measurements to reflect performance
  – Analyze results--What are the metrics indicating about the project
  – Determine project issues

• Evaluate alternative actions to select control changes
  – Requires an evaluation of performance goal resulting from the metrics (EVM)
  – Apply a parametric modeling tool to provide the project expected results

• Implement corrections (Don’t stop measurements!!!)
Control Theory &
Project Management

• Consider a project as a system process which:
  – Follows a detailed plan
  – Can be evaluated in process with process metrics
  – Includes feedback to project management
  – Should incorporate feedback response and control
  – Process control can be applied “during the process” to reduce the risk of not meeting project objectives

• Control systems theory provides a rigorous framework for analyzing complex feedback systems

• Control systems theory can readily incorporate the concepts of optimal control or choosing the best alternative actions to improve performance of the project placed under control
Software Project Management
Fully Managed Project

Desire → Estimating & Planning → Scheduling → Directing → Software Development Process

Measuring
Project Control (Typical)

• Introduce and incorporate measurement process
• Next introduce tools that schedule tasks and allocate resources as some function of inter-task dependencies, resource availability, and priority
• Re-plan
  – Identify all remaining tasks
  – Replan redefined project to begin on the current date with a new best estimate of EAC and schedule to complete
• Problem: Does not consider what may have caused the project to be in trouble in the first place, i.e. size is much larger than expected and/or actual performance is not matching planned performance
• Potential solution: Control Theory incorporate established estimation methodology and algorithms as part of the project evaluation, prediction, and re-planning activities to provide the method to determine the corrective action.
• Performance Based Project Management is a result
Features That Inform and Improve Project Execution

Integration to Project planning tools

Performance Base EVM
Example to demonstrate EVM VS PMC

• Single software element – Mission Management and Control
• Ground based critical system with 498 Weapons Systems standard applied
• Parameters set to K base (expected industry nominal)
• Assumed start date 1-2-2011
• Assumed all new code in C++
• Initial estimate from SEER SEM:
  – 57,607 hours with over 28.76 schedule months
  – End date: 5/25/2013
Baseline Plan from SEER SEM
IMS is assumed to be aligned with SEM estimate

Program : 1.1: Mission planning and control

Baseline Estimate

Development Schedule Months        28.76
Development Effort Hours           57,607
Defect Prediction                  33
Defect Density                     0.83/1000 SLOC
Total Lines Only                   40,000
Effective Lines Only               40,000
SEI Equivalent Rating             3.09
Peak Staff                         19
Effort Probability                 50.00
Schedule Probability               50.00
SW Measurement with EVM and PMC

Results

• System Req Done: Hrs 881 Size 40200 SLOC
  – EVM 57607hrs EAC 0% change 28.7 MO 0% Change
  – PMC 57978hrs EAC 1% change 28.4 MO 0% Change

• SW Req. Done: Hrs 3465 hrs Size 42000 SLOC
  – EVM 57607hrs EAC 0% change 28.7 MO 0% Change
  – PMC 60878hrs EAC 6% change 28.93 MO 0% change

• HLD Done: Hrs 12500 hrs Size 42000 SLOC
  – EVM 57607hrs EAC 0% Change 28.7 MO 0% change
  – PCM 62474hrs EAC 8% change 30.81MO 7%growth

• DD Done: Hrs 25750 hrs Size 42500 SLOC
  – EVM 58825hrs EAC 2% change 30.06MO 5% growth
  – PMC 63088hrs EAC 10% change 32.13MO 12% growth

• NOTE: EVM has no indication of impact of requirements growth impact on EAC until added staff during DD as a result of seeing size growth.
Definitions

**TECHNICAL DEBT**: Effort required to fix violations of good architectural and coding practices that remain in the code when an application is released.

**Violations**: Structure in the source code that is inconsistent with good architectural or coding practices and has proven to cause problems that affect either the cost or risk profile of an application.
Technical Debt (the future cost of defects remaining in code at release, a component of the cost of ownership.

- **Principal** – Cost of fixing problems remaining in the code after release that must be remediated. CRITICAL DEFECTS IN SEER.. Some subset of the structural analysis

- **Interest** - Continuing IT costs attributable to the violations causing technical debt, including higher maintenance costs, greater resource usage, etc. PERFECTIVE IN SEER

- **Liability** —business costs related to outages, breaches, corrupted data, etc. MANUAL COST CALCULATION O BUSINESS LOSSES.. Mean time to defect?? MTTR? Based on amount of tech debt.. Calc bus cust by hour

- **Opportunity cost** —benefits that could have been achieved had resources been put on new capability rather than retiring technical debt.... By doing it right the first time I can save X .... CAN I USE SPEC, TEST, QUAL, MODP.. Experience impacts defects .... Input the defects from cast (use a calibration kbase)
CAST Study on Technical Debt concentrates on 5 Structural Quality Characteristics.

- **Robustness**: The stability of an application and the likelihood of introducing defects when modifying it.
- **Performance**: The efficiency of the software layer of the application.
- **Security**: An application’s ability to prevent unauthorized intrusions.
- **Transferability**: The ease with which a new team can understand the application and quickly become productive working on it.
- **Changeability**: An application’s ability to be easily and quickly modified.
Process To Estimate Cost of Clearing Violations (Perfective Maintenance)

- Determine technology and project type
- Establish a SEER SEM model to represent the project
- Set parameters to match the SEI Level
- Set parameters to tune SEER SEM to specific project, i.e. complexity low if the project is Data Warehouse
- CMMI evaluation to be considered for the project development
- SEER-SEM select perfective maintenance effort for 1 year
- Project size in SLOC or FP
- Apply calibration ratio of CAST database $/1000SLOC to the SEM $/1000 SLOC.

*Note expect the calibration for COBOL applications to be less than 1, expect the calibration for JAVA EE and .NET to be greater than 1.
CAST /SEER Preliminary Example

• Analysis to define a ratio of CAST projected $/KSLOC to SEM projected $/KSLOC
• Ratio developed calibration to perfective maintenance projection
• Validates applying SEER-SEM to provide estimated cost of clearing technical debt
SEER / CAST Sample Projects

• Internal Application project
  – CMMI Level 4
  – Size 3,150KLOC  COBOL
  – Critical violations per KSLOC  0.44
  – Estimate of SEM vs CAST $ per KLOC $1070 vs $1006 (6%)

• Data Warehouse project
  – CMMI Level 4
  – Size 106KLOC  JAVA
  – Estimate of SEM vs CAST $ Per KLOC $828 vs $833 (0.6%)

• Customer Facing WEB application
  – Size 1152 KLOC
  – CMMI Level 3
  – Language JAVA
  – Estimate SEM vs CAST in $/KLOC
    • $4614/$4083 (13%)
Looking At Maintenance and Technical Debt

Base: High Rigor, No Technical Debt

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>Average Effort</th>
<th>Correct</th>
<th>Adapt</th>
<th>Perfect</th>
<th>Total</th>
<th>Cumulative Base Year Cost</th>
<th>Base Year Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>2.9</td>
<td>3.4</td>
<td>10.5</td>
<td>185,034</td>
</tr>
<tr>
<td>2014</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2015</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2016</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2017</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2018</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2019</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2020</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2021</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2022</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>2023</td>
<td>0.0</td>
<td>1.8</td>
<td>2.5</td>
<td>3.0</td>
<td>3.5</td>
<td>10.8</td>
<td>189,709</td>
</tr>
<tr>
<td>TOTALS</td>
<td>18.0</td>
<td>25.5</td>
<td>29.6</td>
<td>34.7</td>
<td>107.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Low Maint. Rigor, Lower Cost, Increased Technical Debt

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>Average Effort</th>
<th>Correct</th>
<th>Adapt</th>
<th>Perfect</th>
<th>Total</th>
<th>Cumulative Base Year Cost</th>
<th>Base Year Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>0.6</td>
<td>3.7</td>
<td>0.4</td>
<td>2.8</td>
<td>3.2</td>
<td>7.2</td>
<td>125,899</td>
</tr>
<tr>
<td>2014</td>
<td>0.6</td>
<td>2.0</td>
<td>0.8</td>
<td>2.3</td>
<td>0.0</td>
<td>6.9</td>
<td>121,720</td>
</tr>
<tr>
<td>2015</td>
<td>0.5</td>
<td>1.1</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,434</td>
</tr>
<tr>
<td>2016</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2017</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2018</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2019</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2020</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2021</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2022</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>2023</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.7</td>
<td>2.0</td>
<td>6.3</td>
<td>110,153</td>
</tr>
<tr>
<td>TOTALS</td>
<td>872.1</td>
<td>13.0</td>
<td>18.9</td>
<td>17.4</td>
<td>921.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

High Actual Cyclomatic Complexity, Low Rigor

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>Average Effort</th>
<th>Correct</th>
<th>Adapt</th>
<th>Perfect</th>
<th>Total</th>
<th>Cumulative Base Year Cost</th>
<th>Base Year Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>18.5</td>
<td>0.4</td>
<td>2.8</td>
<td>0.3</td>
<td>2.1</td>
<td>217.0</td>
<td>3,818,928</td>
</tr>
<tr>
<td>2014</td>
<td>14.4</td>
<td>0.8</td>
<td>2.3</td>
<td>0.6</td>
<td>2.9</td>
<td>190.3</td>
<td>3,050,642</td>
</tr>
<tr>
<td>2015</td>
<td>5.7</td>
<td>63.3</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2016</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2017</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2018</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2019</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2020</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2021</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2022</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>2023</td>
<td>5.5</td>
<td>60.6</td>
<td>1.5</td>
<td>1.7</td>
<td>3.0</td>
<td>65.9</td>
<td>54.5</td>
</tr>
<tr>
<td>TOTALS</td>
<td>872.1</td>
<td>13.0</td>
<td>18.9</td>
<td>17.4</td>
<td>921.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example consideration of Level of Documentation

<table>
<thead>
<tr>
<th></th>
<th>Dev.</th>
<th>MO</th>
<th>Maint MO</th>
<th>Defect Pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>545</td>
<td>53</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Nom</td>
<td>567</td>
<td>49</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Hi</td>
<td>595</td>
<td>46</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Vhi</td>
<td>607</td>
<td>44</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

Low-Min doc. easy recovery from loss
Nom- Mod. loss; Recovery without extreme cost
High- full mil spec with IV&V, Remote Financial tran.
Example consideration of Level of Development Process

<table>
<thead>
<tr>
<th></th>
<th>Dev MO</th>
<th>Maint MO</th>
<th>Defect Pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>567</td>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>Nom</td>
<td>517</td>
<td>36</td>
<td>53</td>
</tr>
<tr>
<td>Hi</td>
<td>472</td>
<td>25</td>
<td>44</td>
</tr>
<tr>
<td>Vhi</td>
<td>432</td>
<td>17</td>
<td>36</td>
</tr>
</tbody>
</table>

Low- Min use of SW development best practices
Nom- reasonable use of some practices
VHI – Routine use of complete SW dev process (CMMI Level 3 or above)
Best Practices Regarding Technical Debt

1. Measure structural quality

2. Establish strategy for retiring technical debt (each or alternate iterations/releases)

3. Prioritize violations for remediation

4. Set thresholds for debt to be retired in each iteration

5. Plan actions in each iteration for retiring technical debt

6. Eliminate causes of technical debt (checklists, training, etc.)
Future Developments in Technical Debt

**Multiple Component Defects (MCDs)**
- Defects that require changes to multiple components
- Require 20X more changes to remediate
- MCDs account for over 50% of remediation effort

**Architectural Hotspots**
- 8%-10% of defects are MCDs
- 20% of components with MCDs contain 80% of MCDs
- These hotspots account for ≈ 40% remediation effort
- Single greatest opportunity to reduce technical debt
References


• Webinar on Technical Debt: http://www.castsoftware.com/resources/technical-debt-estimation

