Software Risk Estimation and Management at JPL

Abstract

The Jet Propulsion Laboratory (JPL) in Pasadena, California is a national laboratory, which is run by the California Institute of Technology for the National Aeronautics and Space Administration (NASA). JPL’s primary roll is to build and operate unmanned, robotic space exploration missions throughout our solar system. JPL, as a Federally Funded Research and Development Center, is always incorporating something new into every software or spacecraft system that it designs and builds. As a result dealing with risk and uncertainty in our estimates has always been a major focus. In the past few years, due to unexpected cost growth on our flight missions and flight software, there has been an increased focus on a more integrated and comprehensive approach to the estimation and management of risk. In this talk we will discuss the following topics related to cost risk:

1. How uncertainty has been incorporated into the JPL software model, probabilistic-based estimates, and how risk is addressed at major milestone reviews since 1989.

2. How cost risk is currently being explored via a variety of approaches, from traditional risk lists, to detailed WBS-based risk estimates to the Defect Detection and Prevention” (DDP) tool. Major issues are arising here as to how to make these approaches work together, as well as how to get them used properly within the JPL environment.

3. Current plans and approach for integrating these different approaches to cost risk and diffusing them into the organization.
Software Risk Estimation and Management at the Jet Propulsion Laboratory

Jairus Hihn
Karen Lum

17th International Forum on COCOMO and Software Cost Modeling
October 22-25, 2002
Background & Context

- NASA’s Jet Propulsion Laboratory is a Federally Funded Research & Development Center whose prime mission is the development and operation of deep space scientific missions.

- JPL has had a very strong emphasis on estimating and managing technical risk for over 40 years.

- Because of hard launch dates schedule was closely managed.

- However, software cost risk has only become a serious focus very recently.
History

- **Pre-1989**
  - Limited use of cost models even though Softcost was originally developed at JPL by R. Tauseworth and D. Reifer. JPL Softcost did not estimate software cost risk.
  - Software cost risk addressed only with risk lists with ‘loosely’ defined mitigation approaches. There was little to no quantification.

- **1989-1996**
  - Developed SCT, a JPL-variant of COCOMO 81 with built-in
    - Monte Carlo algorithms to generate a development effort CDF
    - Calibration
    - Calibration database
    - Used regularly to validate DSN software development effort
  - Software cost risk addressed only with risk lists with ‘loosely’ defined mitigation approaches. There was little to no quantification except when SCT was used.
History continued

- **1996-2001**
  - Software Cost Estimation and Cost Risk activities took a major step backward under Faster, Better, Cheaper
  - Optimistic assumptions were ‘de rigueur’
  - Software cost risk addressed only with risk lists with ‘loosely’ defined mitigation approaches. There was little to no quantification.

- **2001-Today**
  - Software Quality Improvement project and JPL Costing Office Formed
  - Software cost models and formal cost databases required
    - COCOMO II and SEER-SEM
  - Quantitative software cost risk estimates and analysis required
  - JPL Senior Management now ask “Where is your ‘S’ curve?”
  - Numerous explorations into quantitative cost and cost risk management
SW Model Architecture

Figure 1: Overall Cost Quality Modeling Effort

- JPL Software Size Model
- SLOC
- Cost Drivers
- Defect Removal Profiles

Software Project Characteristics

Parametric Cost Models
- COCOMO II
- SEER - SEM
- Price S

Quality Models
- SEER - SEM
- COQUALMO

S.W. Eng. Effort Decomposition

Cost Integrator

Total Effort
Residual Defects
Phase/Activity Cost
Total Dev. Cost
Software Estimation Steps

1. SW Cost Inputs
   - Requirements
   - Architectural Design
   - Mission/Project Sched.
   - Implementation Appr.
   - Mission/Project WBS
   - SW Implementation
   and Design Approach

2. Gather & Analyze
   Technical and
   Programmatic
   Requirements

3. Define Work
   Elements

4. Estimate Software
   Size

5. Estimate Effort

6. Schedule the Effort

7. Calculate Cost

8. Determine the Impact
   of Risk

9. Validate the Estimate
   via Models & Analogy

10. Reconciliation

11. Review & Approve
    Estimates

12. Track & Report
    Estimates

13. Engineering Estimate

14. Estimate Risk

15. Model-Based Estimate

Follow Through
Training Software Managers

Downward bias very likely if estimator does not formally account for underlying probability distribution

Typically cost, effort, SLOC distributions are highly skewed to the right

Point estimates tend to fall between the low and most likely distribution parameters and Most Likely is typically less than 50th percentile

Hihn & Lum
## Uncertainty & Cost Risk Overview

<table>
<thead>
<tr>
<th>Known</th>
<th>Estimate Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>I Forgot’s</td>
<td>Standard WBS Templates &amp; Checklists</td>
</tr>
<tr>
<td>Known Unknowns</td>
<td>Risk Lists</td>
</tr>
<tr>
<td></td>
<td>Quantitative Risk Assessment</td>
</tr>
<tr>
<td>Unknown Unknowns</td>
<td>Design Principle Reserve Percentage</td>
</tr>
</tbody>
</table>
Software Development Cost Cumulative Distribution Function

Recommended Budget with Reserves => 70%
Risk-Adjusted Primary Estimate => 55%, $1.25M
Recommended Minimum without Reserves => 50%
At Primary Estimate => 40% Probability, $1.1M

Cost ($K)

Recommend between $1.2-1.4M
Future Cost Risk Methods

- Estimating 99th percentile and assuming a Log Normal distribution instead of Low, Likely, and High

- Using Cluster analysis to identify analogous projects

- Formal cost risk analysis, mitigation and tracking with DDP
Cost Risk Estimates Based 99th Percentile

Example Cost Risk Sensitivity CDF

- Total
- Mechanical (RMM00)
- Power (RPB00)
- System Engr (R2000)

Cost ($FY'93K)
Cluster Analysis

- Incorporate cluster analysis information to quantify total cost risk
  - Identification of closest analogy should be based on proposal values or similarity to current vintage of estimate
  - Quantitative analysis is focused on history of actual values for analogy mission
DDP Visualizations - Bar Charts

**FM's bar chart**

Unsorted – order matches leaf elements in FM tree

Sorted – in decreasing order of remaining risk

**Green:** of this FM's total Impact on Requirements, that **saved** by PACTs

**Red:** of this FM's total Impact on Requirements, that **remaining** despite PACTs

**Requirements bar chart** – how much each is impacted

**PACTs bar chart** – how much impact each is saving
We may be late bloomers but we are fast learners