Incorporating Risk

Presented by:
Jairus Hihn
JPL Software Quality Improvement Project

International Conference on
Space Mission Challenges for Information Technology
July 13, 2003
Software Estimation Steps

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

SW Cost Inputs

- Applicable Processes & procedures
- Design principles
- Std WBS
- NASA & OMB Reqs

Gather & Analyze
Technical and Programmatic Requirements

Define Work Elements

Engineering Estimate
- Estimate Effort
- Schedule Effort
- Calculate Cost

Rescope

Estimate Software Size

Model-based Estimate

Determine the Impact of Risk

Validation and Reconciliation

Review & Approve Estimates

Track & Report Estimates

Cost Metrics Archive
Incorporating Risk

- The purpose of this step is to identify common software risks, to assess their impact on the cost estimate, and to make revisions to the estimate based on these impacts.

- Risk can be estimated and analyzed in various ways:
  - Expected Risk (Likelihood * Impact)
  - Monte Carlo techniques to capture distributions
  - Cost Models (Previous Lecture)
  - Analyses tools such as ARRT (Advanced Risk Reduction Tool) & DDP (Defect Detection Process)
Background

- This presentation is about cost risk identification and estimation which is only a part of risk management.
- Risk management is an aspect of overall management and includes:
  - cost risk
  - schedule risk (integrated network schedules & critical path)
  - technical risk (good at this but need to map into cost & schedule risk)
- Risk management should be conducted consistently with a risk management process. For example, see:
  - Waltzing with Bears: Managing Risk on Software Projects, Demarco and Lister, 2003
Cost Risk Concepts & Definitions

- Cost Risk is
  - an expression of the uncertainty in work element cost
  - expressed as the
    - probability that a work element will experience cost growth
    - probability distribution of possible final costs for a work element
  - Capturing probabilistic information only requires providing from one to two additional pieces of information
    - Uniform requires a low and a high
    - Triangular requires low, most likely, and high
    - Cost risk table requires a likelihood of occurrence and an impact
# Cost Risk Concepts & Definitions – Sources of Cost Uncertainty

<table>
<thead>
<tr>
<th>Source</th>
<th>How Addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowns</td>
<td>Identify Estimation Uncertainty</td>
</tr>
<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 %</td>
</tr>
</tbody>
</table>
Classic "I Forgot"s

- Review preparation
- Documentation
- Fixing Anomalies and ECR's
- Testing
- Maintenance
- Basic management and coordination activities
  - CogE's do spend time doing management activities
- Mission Support Software Components
- Development and test environments
- Travel
- Training
Cost Risk Concepts, Definitions – Guidelines

- Formal cost risk identifies known unknowns
- Percentage reserve guidelines cover the unknown unknowns
- Risk approach should be simple to understand, use, and track
- Use cost risk estimate to identify reasonable margin
- Flow up to project with cost estimate
- Risk drivers are those events with high probability of occurrence and significant consequence
- Assessing risk at too low a level does not provide any added value
Risk Identification – Generating the Risk List

• As you generate the software significant risk list (SSRL), think about
  – What WBS elements are affected
  – When it would occur
  – Likelihood of occurrence
  – Impact
  – What it would cost to fix it

• Start with project significant risk list (SRL) if exists and common risks

• Develop software level significant risk list (SSRL)
  – Link to specific events for specific task
  – Link to specific WBS elements
  – Link directly to design
  – Identify finite number of “big ticket” items or main risk drivers
Risk Identification – Potential Risks

- Anything New
  - Technology
    - Autonomy
    - Precision landing
    - Hazard avoidance
  - Design
  - Language
  - Tools
  - Development environment
  - Processes
  - Customer or sponsor
- Fixing hardware interface problems
- Concurrent hardware development
- Third party deliverable issues
- Inadequate project definition
- Requirements Volatility
  - Inability to scope flight software due to immature requirements and/or design
- Software Inheritance
- Insufficient technical margins
- Insufficient schedule reserves
- Scope increases
- Cost or effort profile
  - Too low at front end
  - Released too early
- Testbed availability

[These items are based upon causes of cost growth observed at JPL]
Risk Identification – Identifying Main Risk Items

- Systematically go through WBS and identify risk items
- Remember to consider design, SRL, risk check lists

<table>
<thead>
<tr>
<th>WBS Element</th>
<th>Risk Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spacecraft Flight Software</td>
<td></td>
</tr>
<tr>
<td>Software Management</td>
<td></td>
</tr>
<tr>
<td>Software Systems Engineering</td>
<td>Technical margins below Flight Practice Desing Principles</td>
</tr>
<tr>
<td>GN&amp;C</td>
<td>Autonomy</td>
</tr>
<tr>
<td>CT&amp;DM</td>
<td></td>
</tr>
<tr>
<td>Sequencing</td>
<td></td>
</tr>
<tr>
<td>Engineering Applications</td>
<td></td>
</tr>
<tr>
<td>Payload Accomodation</td>
<td>Instrument delivery could be late</td>
</tr>
<tr>
<td>Fault Protection</td>
<td>Current implementation assumes significant SW inheritance</td>
</tr>
<tr>
<td>Software Development Testbed</td>
<td></td>
</tr>
<tr>
<td>Software Integration &amp; Test</td>
<td>Schedule crunch / additional FTEs</td>
</tr>
</tbody>
</table>
Risk Identification
Constructing the SSRL

- Construct SSRL from identified risk items
- Document basic reasons, associated issues, assumptions for identifying each risk item

<table>
<thead>
<tr>
<th>Risk Item</th>
<th>Description</th>
<th>Mitigation Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomy</td>
<td>Existing planner has had some reliability issues that are not fully understood.</td>
<td>Start aggressive prototyping activity immediately to fully identify the issues. In operations could reduce scope of autonomous operations.</td>
</tr>
<tr>
<td>SW Inheritance</td>
<td>Inherited software does not perform as expected.</td>
<td>Current level of inheritance is only 10-15%. Will have to write code from scratch. Can hold inheritance review right after PDR to better determine how much code can really expect to inherit. Set go-nu-go decision point before CDR so can start planning for new code development as early as possible. Can descope to reduce impact.</td>
</tr>
<tr>
<td>Assumptions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low CPU Margins</td>
<td>If CPU margins are too low then do not have sufficient flexibility to handle failures. This can highly constrain design and drives up cost and cost variance significantly.</td>
<td>Oversize system (increase cost) and manage technical margins very carefully. Also carry larger reserves.</td>
</tr>
<tr>
<td>Insufficient I&amp;T</td>
<td>Preceding activities typically over run their schedule putting schedule pressure on I&amp;T. Creates pressure to descope testing activities.</td>
<td>Budget for running multiple shifts.</td>
</tr>
<tr>
<td>Schedule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Late instrument</td>
<td>University XXX has delivered late the last two missions and has not always delivered to specifications.</td>
<td>Hold 1 month fully funded schedule reserve to cover possible code changes and extra testing activities.</td>
</tr>
<tr>
<td>delivery</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7/13/2003              Cost Risk Tutorial                JMH-12
Engineering Cost Risk Estimate

• Complete SSRL matrix
  – For each risk item in the SSRL, estimate
    • Probability of occurrence for each risk
    • Cost impacts if the risk occurs
  – Compute Expected Value (Cost risk for each risk)
  – Calculate the software expected cost risk = Sum of the total expected value in the SSRL
• Without extensive experience and data it can be difficult to generate estimates of the likelihood of occurrence of a risk event

• A standard practice is to pre-set likelihoods for three to five categories. For example,
  – Low 10%
  – Medium 25%
  – High 50%

• You can define these up front as part of your basic assumptions
## Engineering Cost Risk Estimate – SSRL Matrix Example

<table>
<thead>
<tr>
<th></th>
<th>Probability of Occurance (%)</th>
<th>Cost Impact ($K)</th>
<th>Expected Value (PO % * CI $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomy</td>
<td>0.67</td>
<td>$500</td>
<td>$335</td>
</tr>
<tr>
<td>SW Inheritance Assumptions</td>
<td>0.5</td>
<td>$250</td>
<td>$125</td>
</tr>
<tr>
<td>Low CPU Margins</td>
<td>0.25</td>
<td>$500</td>
<td>$125</td>
</tr>
<tr>
<td>Insufficient I&amp;T Schedule</td>
<td>0.67</td>
<td>$125</td>
<td>$84</td>
</tr>
<tr>
<td>Late instrument delivery</td>
<td>0.5</td>
<td>$125</td>
<td>$63</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$731</strong></td>
</tr>
</tbody>
</table>
Engineering Cost Risk Estimate
SSRL Matrix Analysis

- Has the Team thought through potential threats?
  - Have all the risks been identified (Is anything missing?)
  - Do the assessments make sense (Does it pass the “laugh test”?)
    - Impact
    - Likelihood
  - Drive high likelihood risks into budget (>80%)
  - Identify major risk drivers and determine if redesign can lower risk in these areas
  - In example identified risks or known unknowns would be only 9% if budget was $8M. Given cost growth is often 50-100%, need to ask if you have really flushed out all of the potential significant risks.
Using the SSRL Matrix Analysis

- Identify all risk items with likelihood 50% or higher
- Determine risk mitigation strategies for these risks and baseline the mitigation costs into relevant WBS element if mitigation is cheaper then holding cost margin
  - This is where ARRT can be applied
- Else add cost margin to relevant cost elements for identified risk items
- Flow up uncovered SSRL risk items to project
- If budget gets pushed down by manager who does not really understand software, then use risk list and matrix to show impact on risk
  - Also remember: DESCOPE!
Risk Estimation – Process

• Engineering Cost Risk Estimate - Alternate Approach
  – Derive Engineering Cost Risk Estimate by eliciting, for each WBS element, the worst case, most likely and best case cost, then integrating with Monte Carlo methods
Engineering Cost Risk Estimate – Alternate Approach

- Develop cost risk methodology using engineering cost estimate
- Develop risk distributions
  - Cost risk assessment by WBS (cost, schedule, technical & programmatic)
    - Performed at the system, subsystem or component level
  - Determine probability distribution for each WBS element
    - Triangular Distribution: Low, Most Likely (Budget), High
    - Log-normal Distribution: Pessimistic cost either as a Cost or a % of budget
- Run Monte Carlo simulation to combine risk distributions to produce total project cost probability distribution
- Involves subjective expert judgment and/or engineering assessment
## Engineering Cost Risk Estimate – Alternate Approach Inputs

<table>
<thead>
<tr>
<th>Low -10% (SK FY03)</th>
<th>Budget (SK FY03)</th>
<th>High -90% (SK FY03)</th>
<th>Risk Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>899</td>
<td>899</td>
<td>866 Technical margins below Flight Practice Desing</td>
</tr>
<tr>
<td>750</td>
<td>950</td>
<td>1321</td>
<td>1007 Principles</td>
</tr>
<tr>
<td>1850</td>
<td>2761</td>
<td>3367</td>
<td>2659 Autonomy</td>
</tr>
<tr>
<td>1350</td>
<td>1492</td>
<td>1959</td>
<td>1600</td>
</tr>
<tr>
<td>500</td>
<td>543</td>
<td>600</td>
<td>548</td>
</tr>
<tr>
<td>275</td>
<td>298</td>
<td>350</td>
<td>308</td>
</tr>
<tr>
<td>200</td>
<td>275</td>
<td>300</td>
<td>258 Instrument delivery could be late</td>
</tr>
<tr>
<td>750</td>
<td>858</td>
<td>1206</td>
<td>938 SW inheritance</td>
</tr>
<tr>
<td>50</td>
<td>75</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>100</td>
<td>125</td>
<td>175</td>
<td>133 Schedule crunch / additional FTEs</td>
</tr>
</tbody>
</table>

7/13/2003

Cost Risk Tutorial

JMH-20
Reporting Subsystem Risk

- Reevaluate and re-examine the SSRL and adjust the SSRL matrix, if required
- Examine other areas for subsystem related risks
  - Critical path activities
  - Long lead items
  - Supplier past performance record
  - Availability of people and facilities
- Finally PEM or CogE flows up the augmented SSRL matrix
Wrap UP

• Estimating the cost risk enables the PEM or CogE to
  – Identify reasonable margin
  – Identify when mitigation actions are needed
  – Be able to show quickly what is changing when budgets get pushed down. Lower budgets mean higher risk and decreased scope

• Main outputs of this activity
  – Risk adjusted cost estimate
  – SSRL and Matrix
  – Mitigation actions
Appendix
# Software Cost Risk Drivers and Ratings

<table>
<thead>
<tr>
<th>Risk Drivers</th>
<th>Software Cost Risk Driver Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Nominal (Reduces Risk)</strong></td>
</tr>
<tr>
<td>Experience &amp; Teaming</td>
<td>• Extensive software experience in the project office</td>
</tr>
<tr>
<td></td>
<td>• Software staff included in early planning and design decisions</td>
</tr>
<tr>
<td></td>
<td>• Integrated HW and SW teams</td>
</tr>
<tr>
<td>Planning</td>
<td>• Appropriately detailed and reviewed Plan</td>
</tr>
<tr>
<td></td>
<td>• All key parties provide input with time to get buy-in</td>
</tr>
<tr>
<td></td>
<td>• Appropriate assignment of reserves</td>
</tr>
<tr>
<td></td>
<td>• SW inheritance verified based on review and adequate support</td>
</tr>
<tr>
<td>Requirements &amp; Design</td>
<td>• Solid system and SW architecture with clear rules for system partitioning</td>
</tr>
<tr>
<td></td>
<td>• Integrated systems decisions based on both HW and SW criteria</td>
</tr>
<tr>
<td></td>
<td>• SW Development process designed to allow for evolving requirements</td>
</tr>
<tr>
<td>Staffing</td>
<td>• Expected turnover is low</td>
</tr>
<tr>
<td></td>
<td>• Bring software staff on in timely fashion</td>
</tr>
<tr>
<td></td>
<td>• Plan to keep software team in place through launch</td>
</tr>
<tr>
<td>Testing</td>
<td>• Multiple Test-beds identified as planned deliverables and scheduled for early completion.</td>
</tr>
<tr>
<td></td>
<td>• Separate test team</td>
</tr>
<tr>
<td></td>
<td>• Early development of test plan</td>
</tr>
<tr>
<td>Tools</td>
<td>• CM and Test tools appropriate to project needs</td>
</tr>
<tr>
<td></td>
<td>• Proven design tools</td>
</tr>
</tbody>
</table>

7/13/2003 Cost Risk Tutorial JMH-25
## Estimated Cost Impact of Risk Drivers

<table>
<thead>
<tr>
<th>Risk Drivers</th>
<th>Estimated Cost Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Experience &amp; Teaming</td>
<td>1.02</td>
</tr>
<tr>
<td>Planning</td>
<td>1.10</td>
</tr>
<tr>
<td>Requirements &amp; Design</td>
<td>1.05</td>
</tr>
<tr>
<td>Staffing</td>
<td>1.02</td>
</tr>
<tr>
<td>Testing</td>
<td>1.05</td>
</tr>
<tr>
<td>Tools</td>
<td>1.02</td>
</tr>
<tr>
<td>Maximum Expected Cost Impact</td>
<td>1.30</td>
</tr>
</tbody>
</table>
Rules-of-Thumb

• JPL-Based "Rules-of-Thumb":
  – The six risk drivers, in the Tables 11 and 12 were identified based on a study of seven JPL missions that experienced significant cost growth [Hihn and Habib-agahi, 2000]
  – 80% to 100% of attempts to inherit software not written for inheritance fails [Hihn and Habib-agahi, May 2000, Hihn and Habib-agahi, Sept. 2000]
Rules of Thumb (2)

• "Rules-of-Thumb" from other Sources:
  - 55% of software projects exceed budget by at least 90%.
    • Software projects at large companies are not completed 91% of the time
    • Of the projects that are completed, only 42% of them have all the originally proposed features [Remer, 1998]
  - Historical cost estimates for NASA projects are underestimated by a factor of at least 2
    • The actual versus estimated cost ratio is from 2.1 to 2.5 [Remer, 1998]
  - Cost estimation accuracy using ratio estimating by phases without detailed engineering data gives an accuracy of −3% to +50%
    • Using flow diagram layouts, interface details, etc. gives an accuracy of −15% to +15%
    • Using well defined engineering data, and a complete set of requirements gives an accuracy of −5% to +15% [Remer, 1998]
Rules-of-Thumb (3)

- An accuracy rate of -10% to +10% requires that 7% of a rough order of magnitude budget and schedule be used to develop the plan and budget
  - Another way to look at this is to consider the percentage of total job calendar time required
  - When using existing technology, 8% of calendar/budget should be allocated to plan development
  - When high technology is used, then 18% of calendar/budget should be allocated to plan development [Remer, 1998]

- According to Boehm [Boehm, et. al., 2000], the impacts of certain risk drivers can be significantly higher than the JPL study:
  - Requirements volatility can increase cost by as much as 62%
  - Concurrent hardware platform development can increase cost by as much as 30%
  - Incorporating anything for the first time, such as new design methods, languages, tools, processes can increase cost by as much as 20%, and if there are multiple sources of newness, it can increase cost as much as 100%