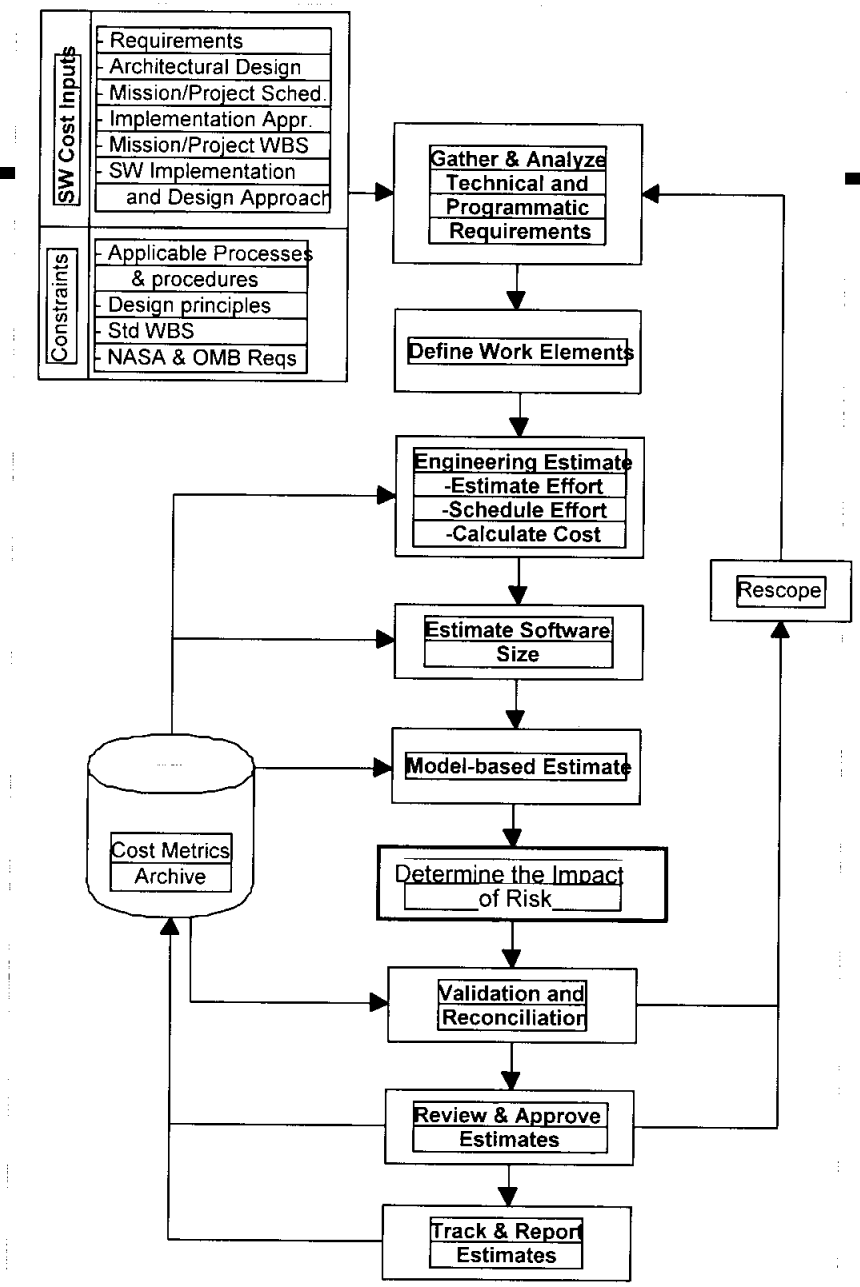
Incorporating Risk

Presented by:
Jairus Hihn
JPL Software Quality Improvement Project

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

Software Estimation Steps



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

Source	How Addressed
Knowns	Identify Estimation Uncertainty
"I Forgot"s	Standard WBS Templates & Checklists
Known Unknowns	Risk Lists Quantitative Risk Assessment
Unknown Unknowns	Design Principle Reserve %

Best Practices

Focus of Cost Risk Estimation

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
- Insufficent 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

WBS Element	Risk Item
Spacecraft Flight Software	
Software Management	
Software Systems Engineering	Technical margins below Flight Practice Desing Principles
GN&C	Autonomy
CT&DM	
Sequencing	
Engineering Appplications	
Payload Accomodation	Instrument delivery could be late
Fault Protection	Current implementation assumes siginificant SW inheriteance
Software Development Testbed	
Software Integration & Test	Schedule crunch / additional FTEs

Risk Identification Constructing the SSRL

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

Risk Item	Description	Mitigation Action
Autonomy	Existing planner has had some reliability issues that are not fully understood.	Start aggressive prototyping activity immediately to fully identify the issues. In operations could reduce scope of autonomous operations.
SW Inheritance Assumptions	Inherited software does not perform as expected.	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-no-go decision point before CDR so can start planning for new code development as early as possible. Can descope to reduce impact.
Low CPU Margins	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.	Oversize system (increase cost) and manage technical margins very carefully. Also carry larger reserves.
Insufficient I&T Schedule	Preceding activities typically over run their schedule putting schedule pressure on I&T. Creates pressure to descope testing activities.	Budget for running multiple shifts.
Late instrument delivery	University XXX has delivered late the last two missions and has not always delivered to specificiations.	Hold 1 month fully funded schedule reserve to cover possible code changes and extra testing activities.

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

Guidance for Generating Numbers

- 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

	Probability of Occurance (%)	Cost Impact ($K)	Expected Value (PO % * CI $)
Autonomy	0.67	$500	$335
SW Inheritance Assumptions	0.5	$250	$125
Low CPU Margins	0.25	$500	$125
Insufficient I&T Schedule	0.67	$125	$84
Late instrument delivery	0.5	$125	$63
Total			$731

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

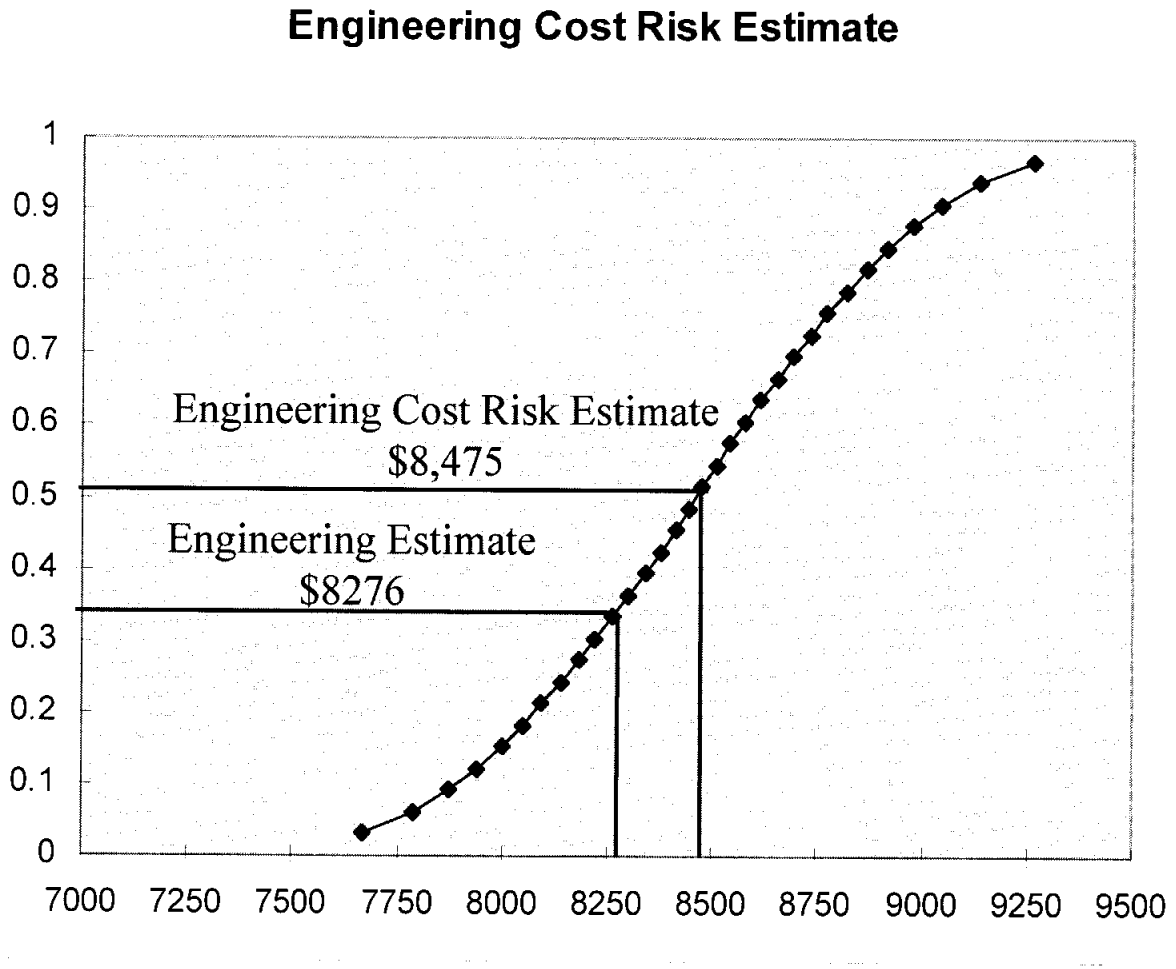
Engineering Cost Risk Estimate – Alternate Approach

- 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

Low -10% ($K FY03)	Budget ($K FY03)	High -90% ($K FY03)		Risk Item
800	899	899	866	
750	950	1321	1007	Technical margins below Flight Practice Desing Principles
1850	2761	3367	2659	Autonomy
1350	1492	1959	1600	
500	543	600	548	
275	298	350	308	
200	275	300	258	Instrument delivery could be late
750	858	1206	938	SW inheriteance
50	75	100	75	
100	125	175	133	Schedule crunch / additional FTEs
6625	8276	10277	8393	

Engineering Cost Risk Estimate – Alternate Approach Outputs



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

Risk Drivers	Software Cost Risk Driver Ratings	
	Nominal (Reduces Risk)	**Extra High (Increases Risk)**
Experience & Teaming	• Extensive software experience in the project office • Software staff included in early planning and design decisions • Integrated HW and SW teams	• Limited software experience in the project office • Software staff not included in early planning and design decisions • HW and SW teams are not integrated
Planning	• Appropriately detailed and reviewed Plan • All key parties provide input with time to get buy-in • Appropriate assignment of reserves • SW inheritance verified based on review and adequate support	• Lack of appropriate planning detail with insufficient review • Not all parties involved in plan development • Simplistic approach to reserve allocation • Optimistic non-verified assumptions especially with respect to software inheritance
Requirements & Design	• Solid system and SW architecture with clear rules for system partitioning • Integrated systems decisions based on both HW and SW criteria • SW Development process designed to allow for evolving requirements	• System and Software architecture not in place early with unclear descriptions of basis for HW & SW partitioning of functionality. • Systems decisions made without accounting for impact on software • Expect SW requirements to solidify late in the life-cycle
Staffing	• Expected turnover is low • Bring software staff on in timely fashion • Plan to keep software team in place through launch	• Expected turnover is high • Staff up software late in life-cycle • Plan to release software team before ATLO
Testing	• Multiple Test-beds identified as planned deliverables and scheduled for early completion. • Separate test team • Early development of test plan	• Insufficient Test-beds/simulators dedicated to SW and are not clearly identified as project deliverables • Plan to convert SW developers into test team late in life-cycle • Test documents not due till very late in the life-cycle
Tools	• CM and Test tools appropriate to project needs • Proven design tools	• No or limited capability CM and test analysis tools • Unproven design tools selected with limited time for analysis

Estimated Cost Impact of Risk Drivers

Risk Drivers	Estimated Cost Impact		
	High	Very High	Extra High
Experience & Teaming	1.02	1.05	1.08
Planning	1.10	1.17	1.25
Requirements & Design	1.05	1.13	1.20
Staffing	1.02	1.05	1.13
Testing	1.05	1.08	1.15
Tools	1.02	1.03	1.10
Maximum Expected Cost Impact	1.30	1.60	2.32

Rules-of-Thumb (1)

- 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%