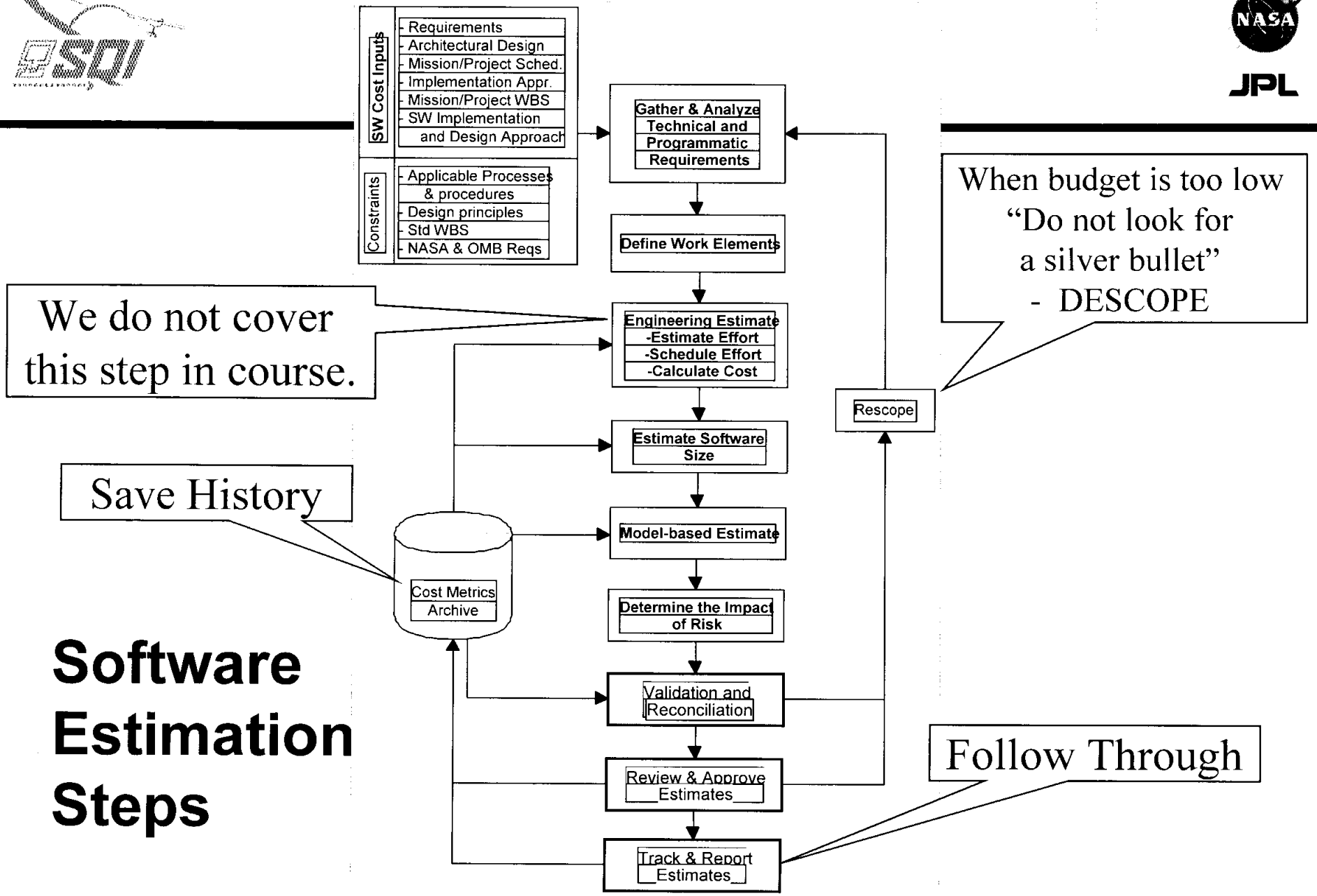
Reconciliation,Closing the Deal and Following Through

Presented by:
Jairus Hihn
JPL Software Quality Improvement Project

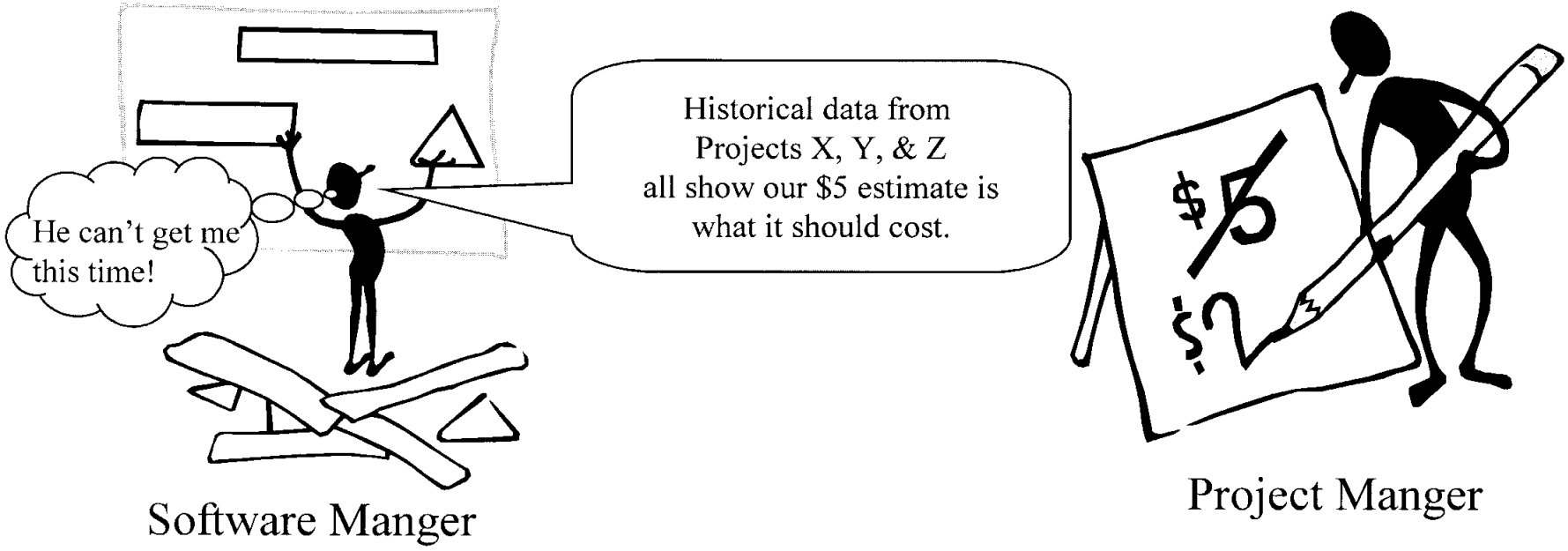
International Conference on
Space Mission Challenges for Information Technology
July 13, 2003

Software Estimation Steps

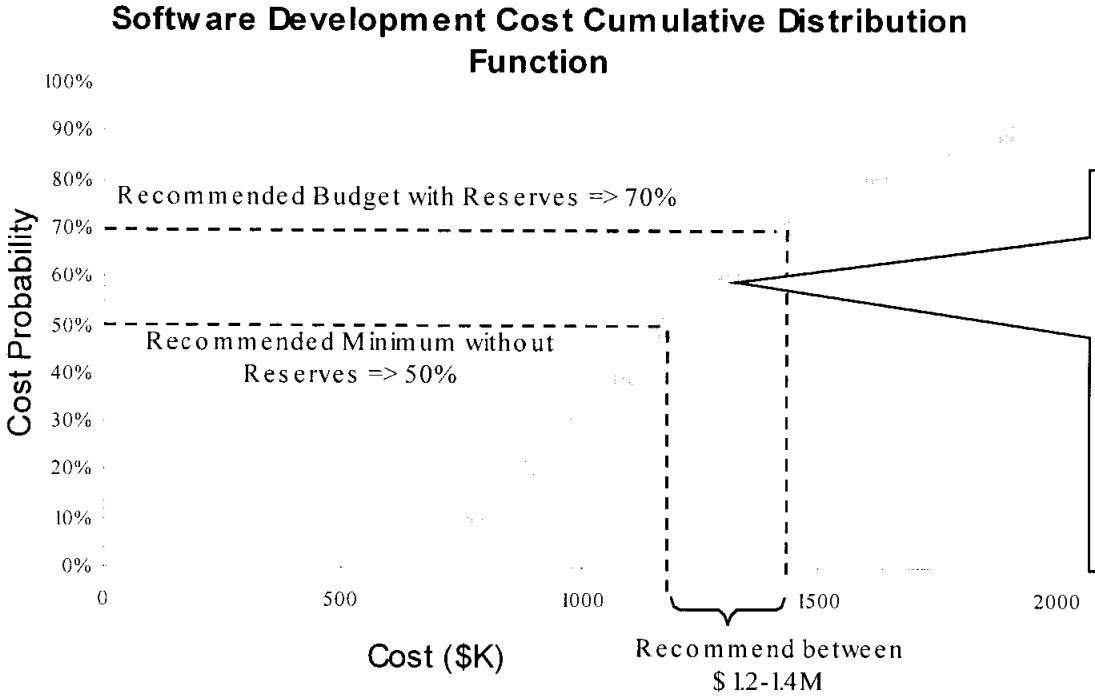


Validate & Reconcile The Estimate

- The purpose of this step is to validate and reconcile the estimates and improve their accuracy via comparing the risk adjusted engineering estimate with
  - Model estimates
  - Analogies (use several if available)
- This provides a more rigorous BOE and improves estimate credibility

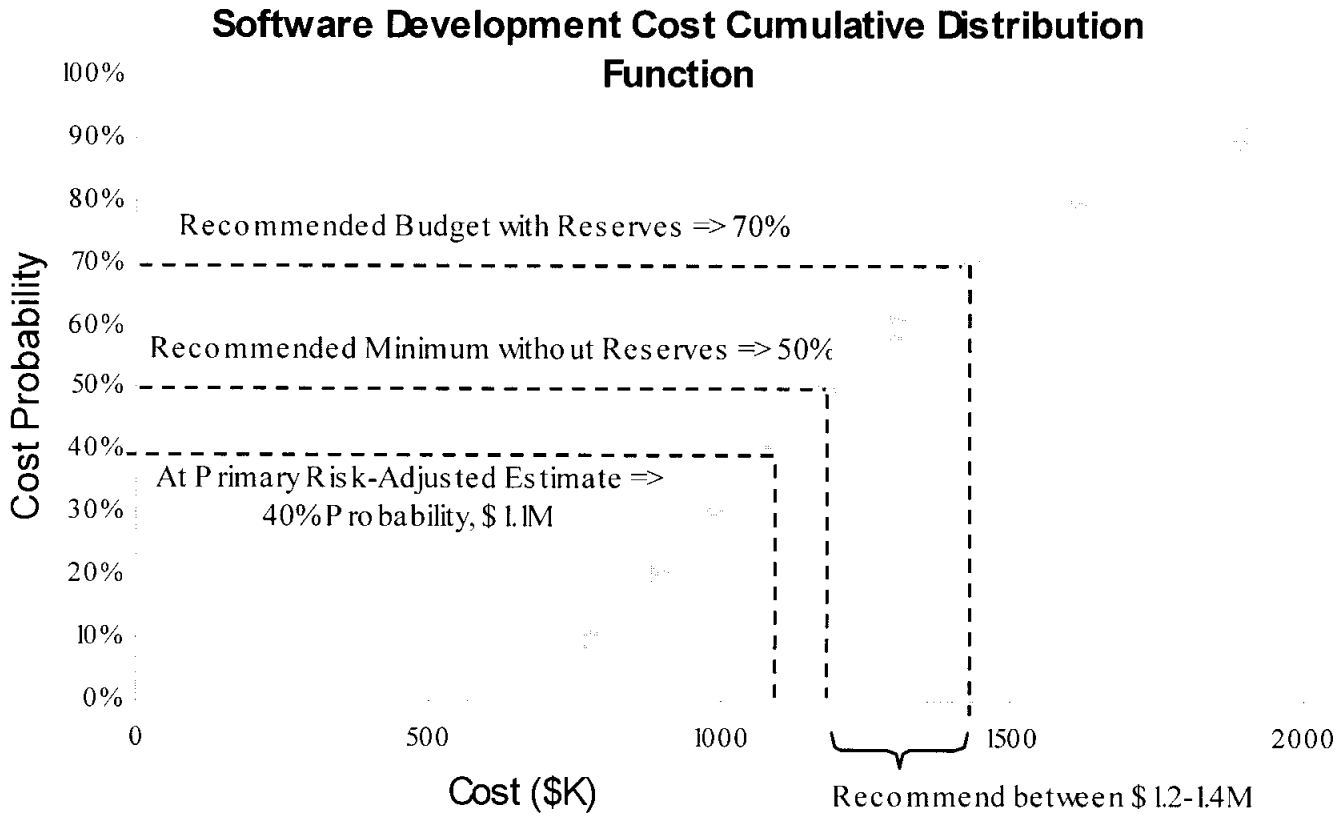


Example Model Output

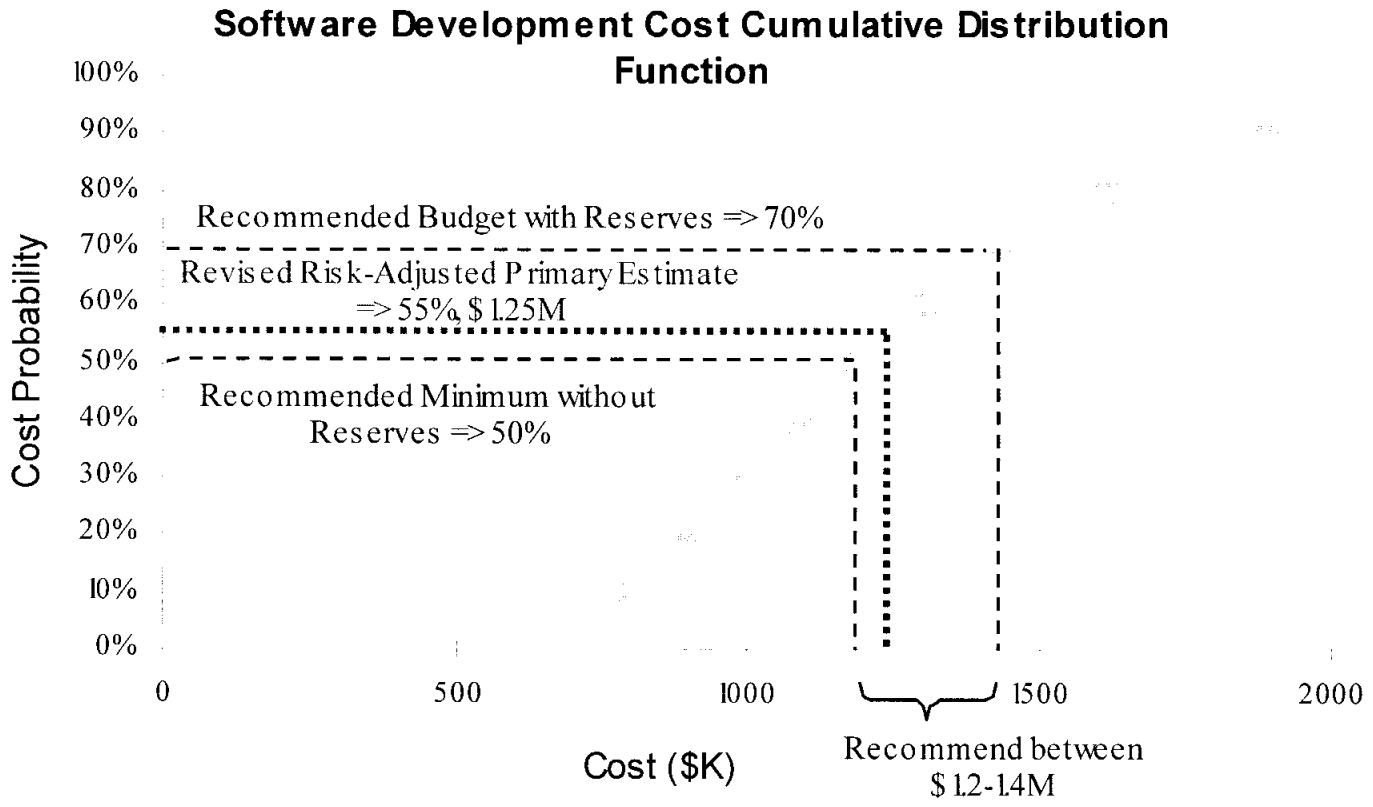


- For tasks with 10% level of reserves or less recommend a range of 50% to 70% probability
- For tasks with 20% or greater reserves recommend 40-65% for other Subsystems

Inconsistent Estimates Example



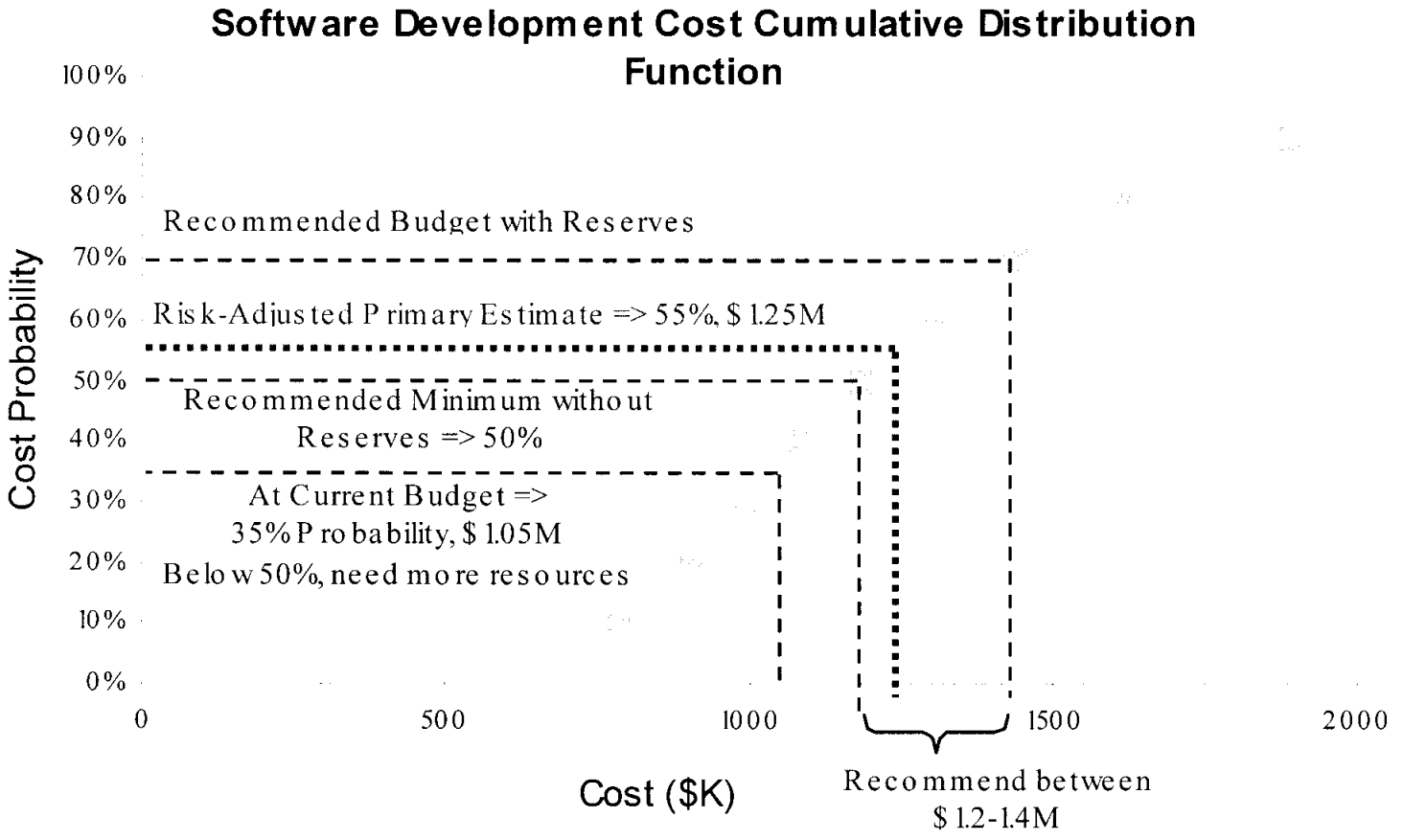
Consistent Estimates Example



Reconciling the Risk Estimates

- When the engineering cost risk estimate is below 50%, iterate and rework estimates to determine cause of disparity
  - Check assumptions
  - Check technical margins
  - Check design drivers
  - Double check "I Forgot"s
  - Review schedule
- When the engineering cost risk estimate is greater then 70%, Iterate and rework estimates to determine cause of disparity
  - Analyze risk drivers to verify risk assumptions
  - Review design & approach for changes to reduce risk
  - Check model assumptions
- If your primary estimate is below 50%, your primary estimate and model-based estimate should be examined for:
  - overly pessimistic or optimistic assumptions.
  - any forgotten items
- Iterate this step until your estimates are consistent or differences can be justified

Validation of Budget Example



Review, Document & Archive the Estimate

- The purpose of this step is to
  - Review the software estimates
  - Obtain project and line management approval.
  - Check the accuracy of the software estimates over time
  - Maintain your history
- This is the step where things can fall apart
- Your lines of defense
  - Need defendable estimate
  - Descope

Review Objectives

- Confirm the WBS and the software architecture
- Verify the methods used for deriving the size, effort, schedule, and cost
- Ensure the BOE, assumptions and input data used to develop the estimates are correct
- Ensure that the estimates are reasonable and accurate, given the input data

Things to Look Out For

- Current proposal and planning process encourages/ demands under-estimating in early stages of lifecycle
- Optimistic assumptions with respect to
  - Ability to reuse/inherit existing software
  - Availability of new technology and products
  - Stability and understanding of requirements
  - Newness
    - Anything you do for the first time will cost more than you expect
- Classic "I Forgot"s
  - Review preparation
  - Documentation
  - Anomaly and ECR's
  - Testing
  - Maintenance
  - Basic management and coordination activities
  - Mission Support Software Components

When In Doubt…

DESCOPE!!!

Track Your History

- Track the estimates to identify when, how much, and why the project may be overrunning or under-running the estimates.
  - Compare current estimates, and ultimately actual data, with past estimates and budgets to determine the variation of the estimates over time
  - This allows estimators to see how well they are estimating and how the software project is changing over time
- Track performance to calibrate your team

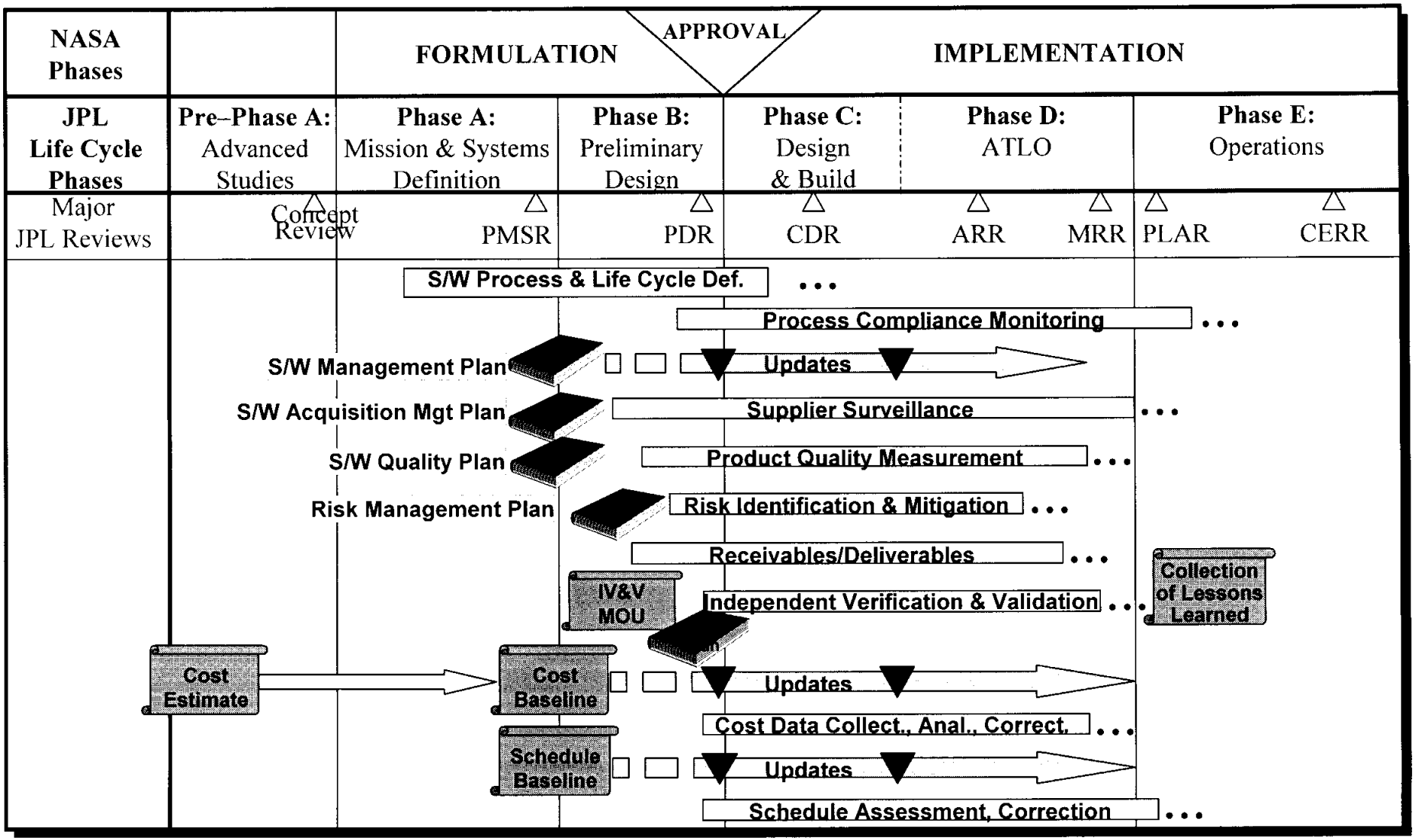
Estimate Data to Archive

- Project characteristics (project name, software organization, software type, application domain, platform, language, and mission class)
- Date of approved estimate
- Planned work breakdown structure (WBS)
- Estimated effort and cost of each work element of the WBS
- Estimation assumptions made (rules-of-thumb, development productivity, etc.)
- Estimation methods used
- Estimated size, effort and cost of software functions at the lowest level possible (subsystem, object, etc.)
- Estimated total size, effort and cost
- Planned dates of major milestones and reviews from the schedule
- Planned procurements
- Estimated cost of procurements
- Identified risks and their estimated likelihood & impact

Completion Data to Archive

- Project characteristics (project name, software organization, software type, application domain, platform, language, and mission class)
- Date of completion of work elements
- Actual work breakdown structure
- Actual development productivity
- Actual size of each software function at the lowest level possible (subsystem, object, etc.), in SLOC
- If available, actual effort and cost of software functions at the lowest level possible (subsystem, object, etc.)
- If available, actual effort and cost of each work element of the WBS
- Actual total size, effort and cost
- Actual dates of major milestones and reviews from the schedule
- Actual procurements
- Actual cost of procurements
- If available, actual risks and their impact

Software Cost Estimation & Planning Over the Life-Cycle



Last But Not Least

- Don't let precision carry you away to accuracy
- Maintain good communication with customer(s) and cognizant engineers to minimize surprises and maximize chances of meeting customer needs

DESCOPE!!!