



JPL



Software Cost 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**



■ 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

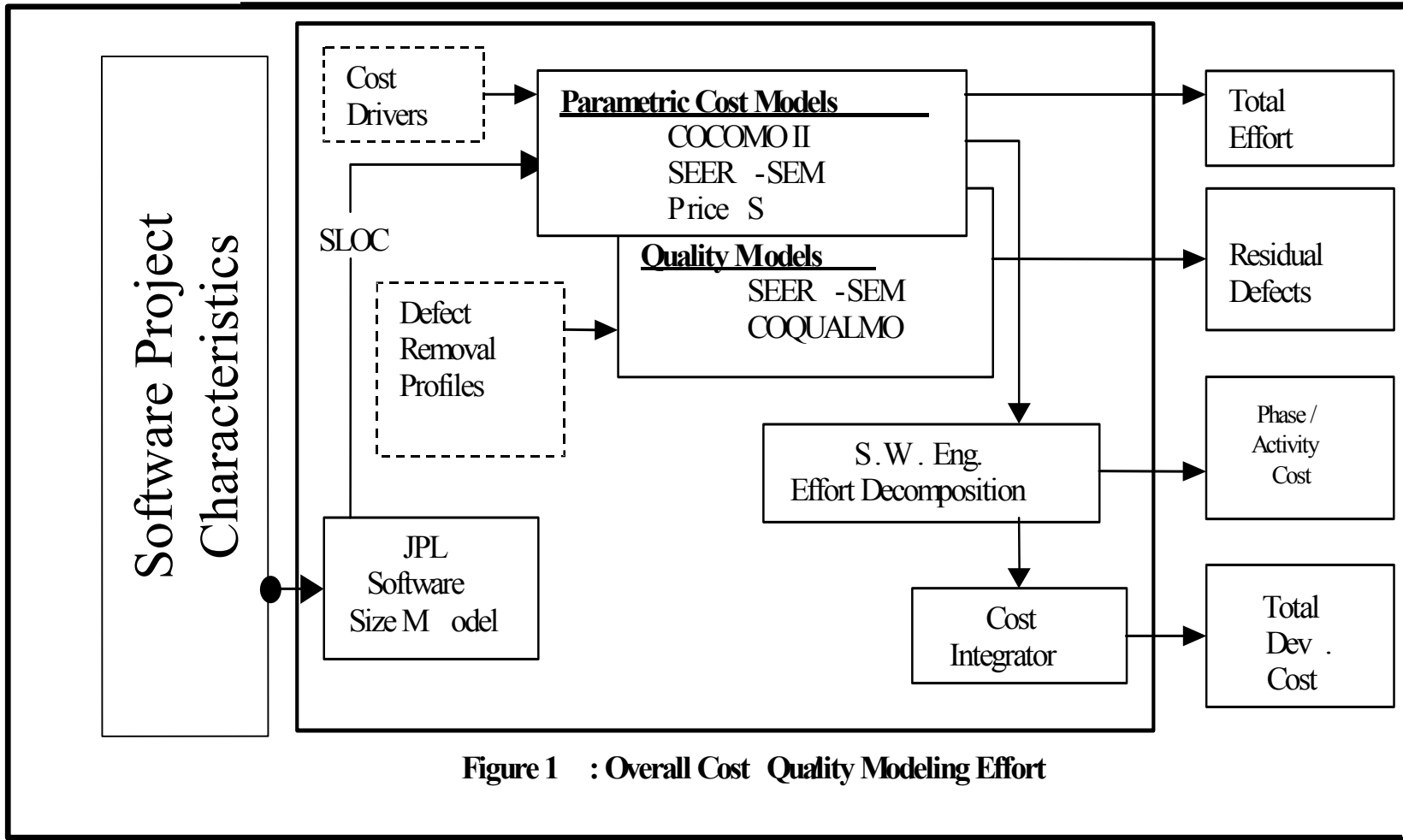
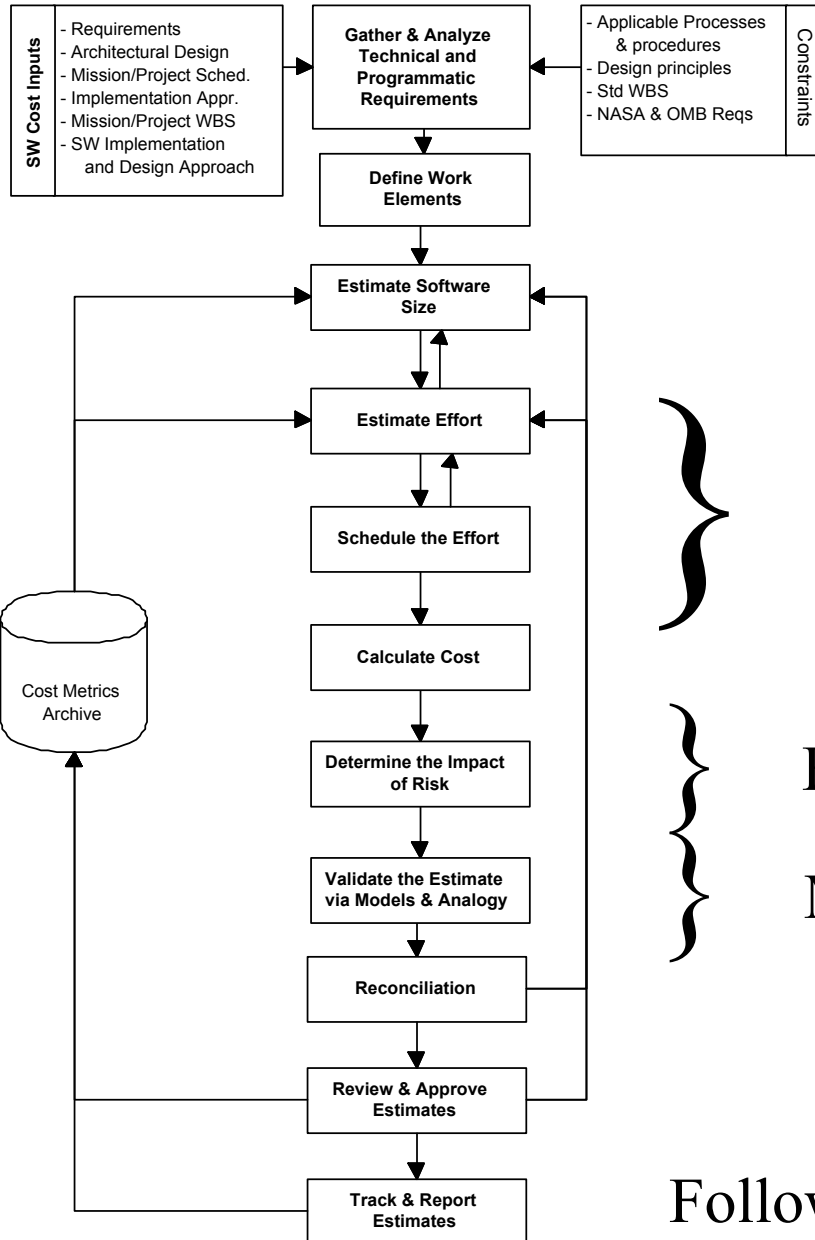


Figure 1 : Overall Cost Quality Modeling Effort

Software Estimation Steps



Save History

Engineering Estimate

Estimate Risk

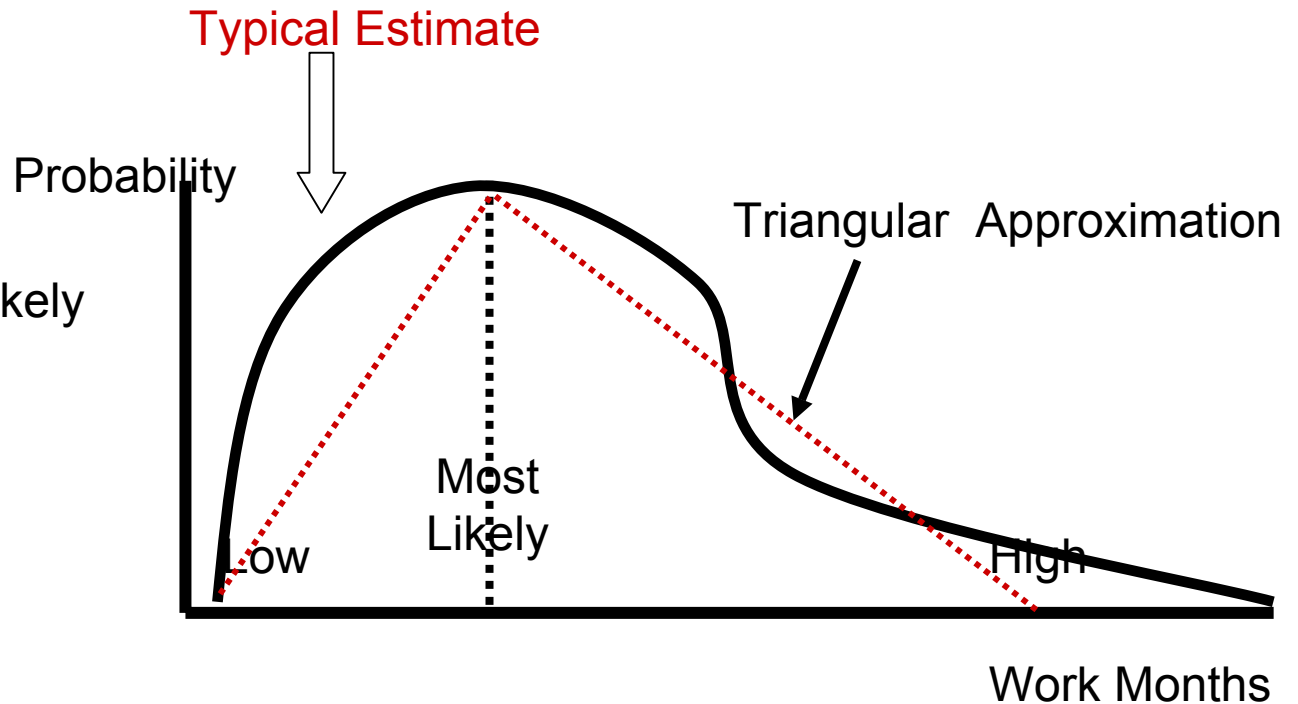
Model-Based Estimate

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

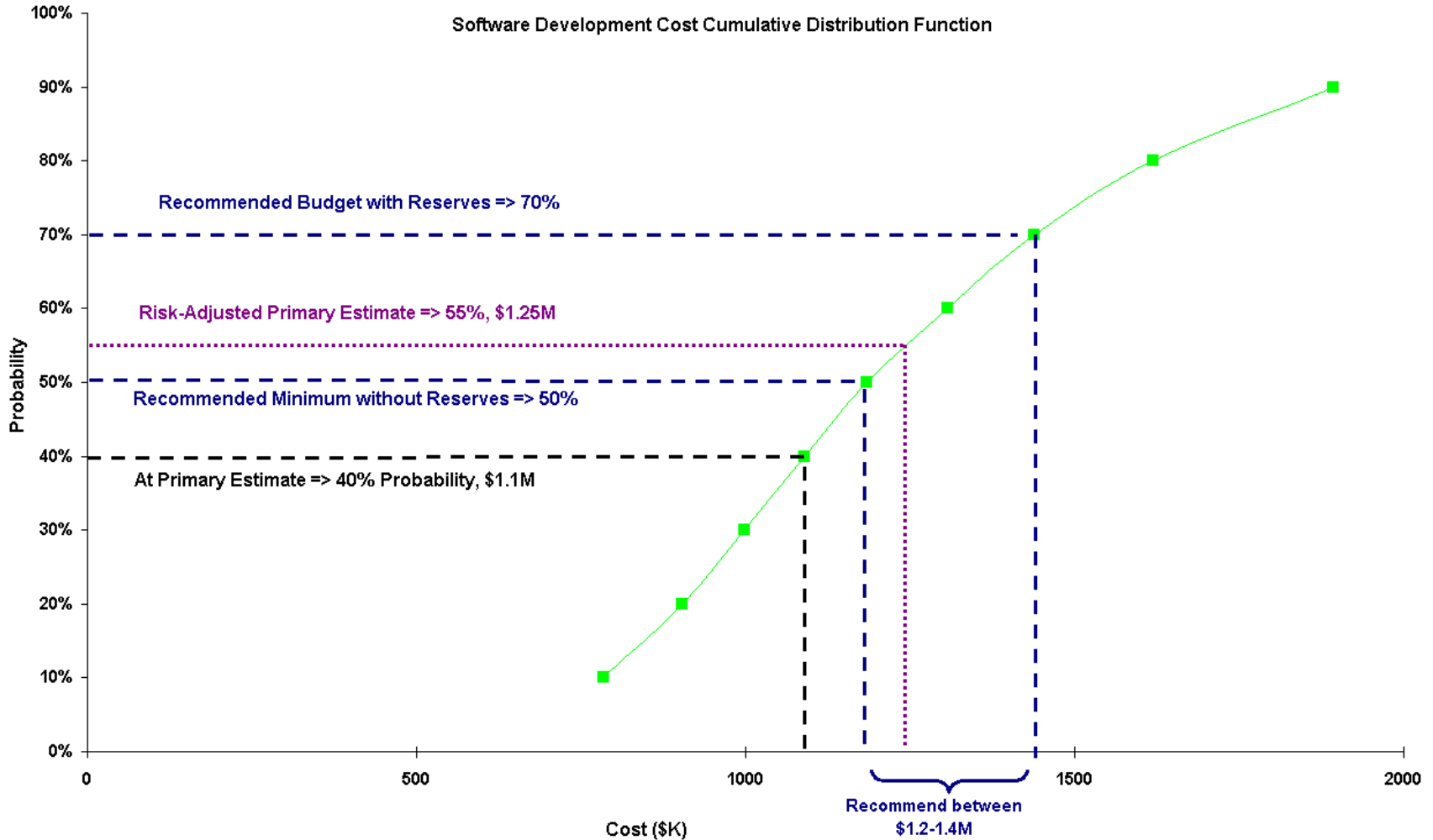
Uncertainty & Cost Risk Overview



Known	Estimate Uncertainty
I Forgot's	Standard WBS Templates & Checklists
Known Unknowns	Risk Lists Quantitative Risk Assessment
Unknown Unknowns	Design Principle Reserve Percentage



Standard JPL SW Cost Risk Estimate



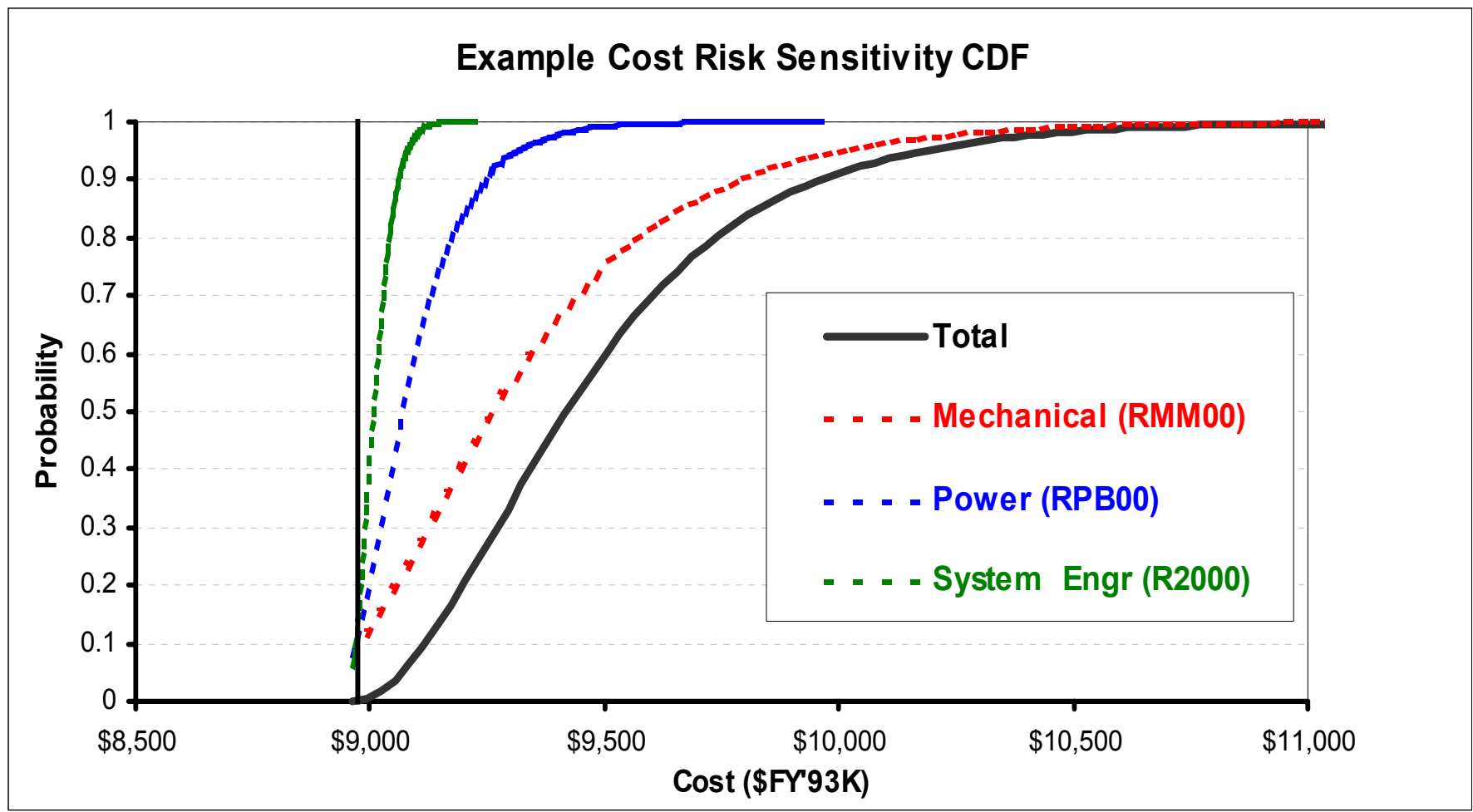
Future Cost Risk Methods

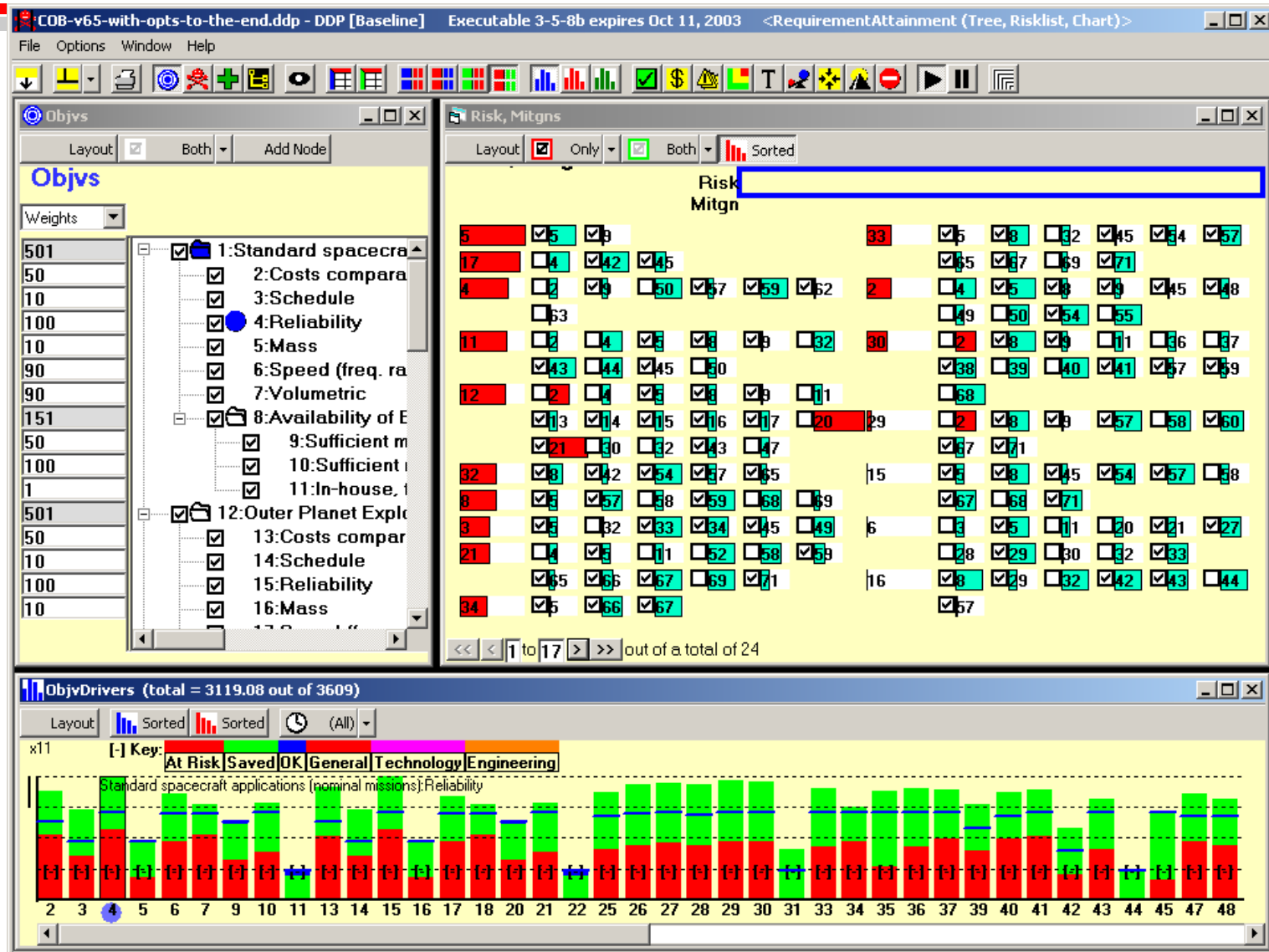


- **Estimating 99th percentile and assuming a Log Normal distribution instead of Low, Likely, and High**
- **Using Cluster analysis to identify analogous projects**
- **Formal cost risk analysis, mitigation and tracking with DDP**



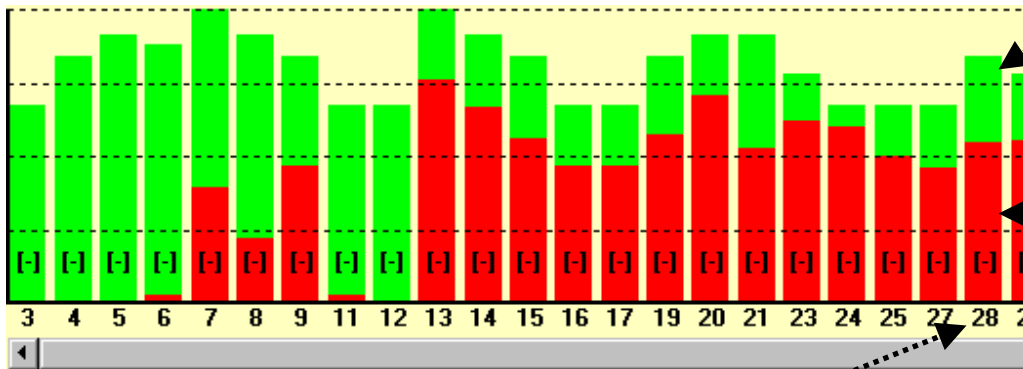
Cost Risk Estimates Based 99th Percentile





FMs bar chart

Unsorted – order matches leaf elements in FM tree

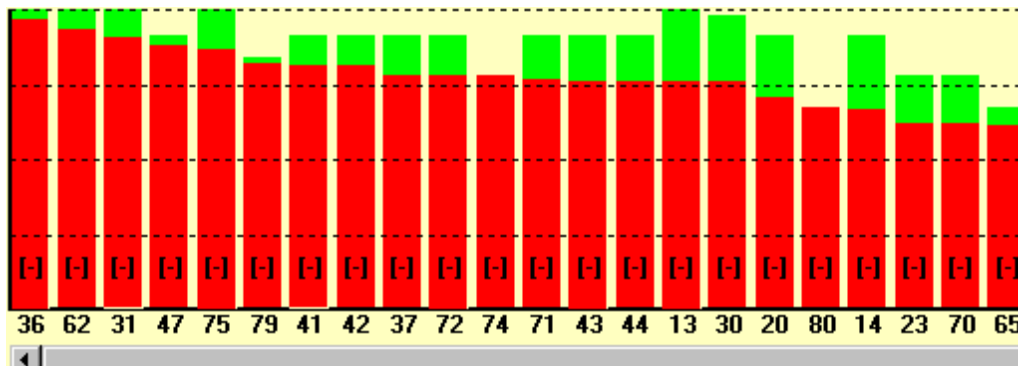


Item number in tree

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

Sorted – in decreasing order of remaining risk



Requirements bar chart – how much each is impacted

PACTs bar chart – how much impact each is saving

Conclusion



**We may be late bloomers
but we are fast learners**