



# SMC-IT Cost Risk Tutorial

---



## Introduction

Presented by:  
Jairus Hihn

JPL Software Quality Improvement Project

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



# Topics



- Purpose/Objective of Class
- Scope of Material
- Estimation Basics
  - Estimation Methods
  - Cost Uncertainty
- Overview Of Software Estimation Steps



# Agenda



---

Lecture 1: INTRODUCTION	08:30
Lecture 2: SCOPING THE JOB	09:00
Lecture 3: SIZING THE SYSTEM	09:30
EXERCISE	
Lecture 4: MODEL BASED ESTIMATE	10:45
EXERCISE	
Lecture 5: INCORPORATING RISK (Begin)	12:00
Lunch	12:30
Lecture 5: INCORPORATING RISK (Finish)	14:00
Lecture 6: VALIDATING, RECONCILING, AND REVIEWING	14:30
Lecture 7: USING ARRT	15:00



# Class Purpose/Objective



- Purpose is to provide a structured tutorial on using a set of tools processes and methods for estimating and analyzing cost risk
- Objective is that by the end of day you should be able to make an estimate using the described process and methods, and available tools and data
- Supplemental materials provided
  - *Handbook for Software Cost Estimation*, D-26303
  - Excel based tools
    - Software Cost Analysis Tool – probabilistic COCOMO II
      - User Guide
    - Monte Carlo Excel add-ins
  - Advanced Risk R Tool



# Scope (1)



- Can be used by anyone who has to make a software cost estimate. But main focus is on
  - Software Managers
  - Software Engineers
- The estimation method emphasizes development of cost estimates based upon:
  - Data-driven estimates from JPL historical experience
  - Risk and uncertainty impacts on estimates
  - Multiple estimates



# Scope (2)



- For flight software, estimate includes activities up to launch, and for ground software, this usually consists of activities up to deployment:

*requirements analysis*

*design*

*coding*

*software integration and test*

*system test*

- Software development cost for new, reused and modified code
- Activities included are :

*software management*

*configuration management*

*software quality assurance*

*software development*

- As well as other costs that must also be considered when developing an overall cost estimate:

*hardware procurement costs*   *travel costs*



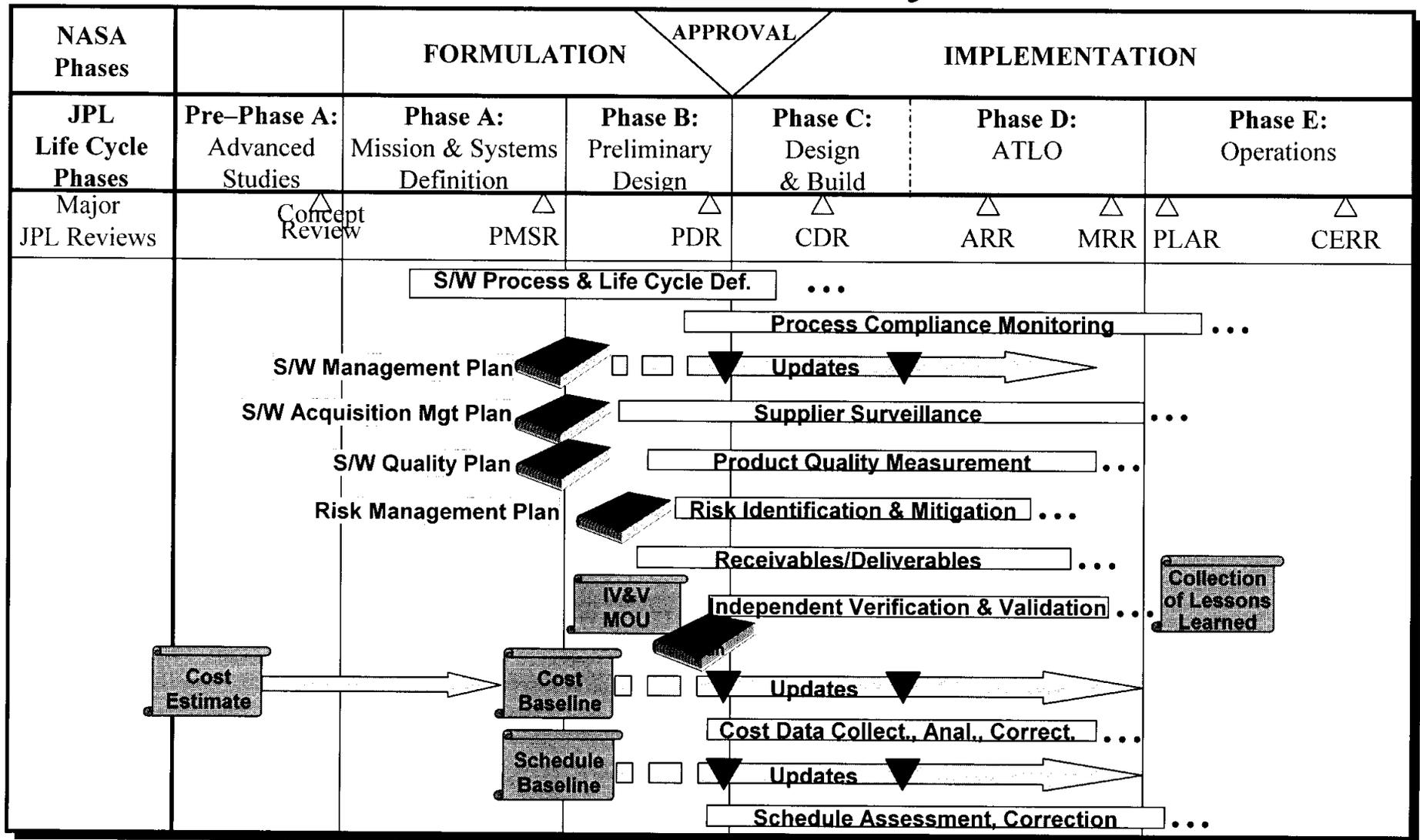
# Cost Estimation is NOT an Isolated Activity

---



- Cost estimation should never be an activity that is performed independently
- In the early life-cycle phases, cost estimation is closely related to design activities, where the interaction between these activities is iterated many times as part of doing design trade studies and early risk analysis
- Current proposal and planning process encourages/ demands under-estimating in early stages of lifecycle
- Later on in the life-cycle, cost estimation supports management activities – primarily detailed planning, scheduling, and risk management
  - You will get held to this number!!

# Software Cost Estimation & Planning Over the Life-Cycle





# Estimation Methods



- All estimates are made using some combination of the basic estimation methods
  - Analogy
    - formal analogy
    - expert judgment
  - Cost estimating Relationship (CER)
    - models
    - “rules-of-thumb.”
- Whatever method is used, it is most important that the assumptions and formulas used be documented to enable more thorough reviews, and to make it easier to revise estimates at future dates when assumptions may need to be revised. Documented estimate is first line of defense against arbitrary budget cuts.



# Software Estimation Is An Uncertain Business

---



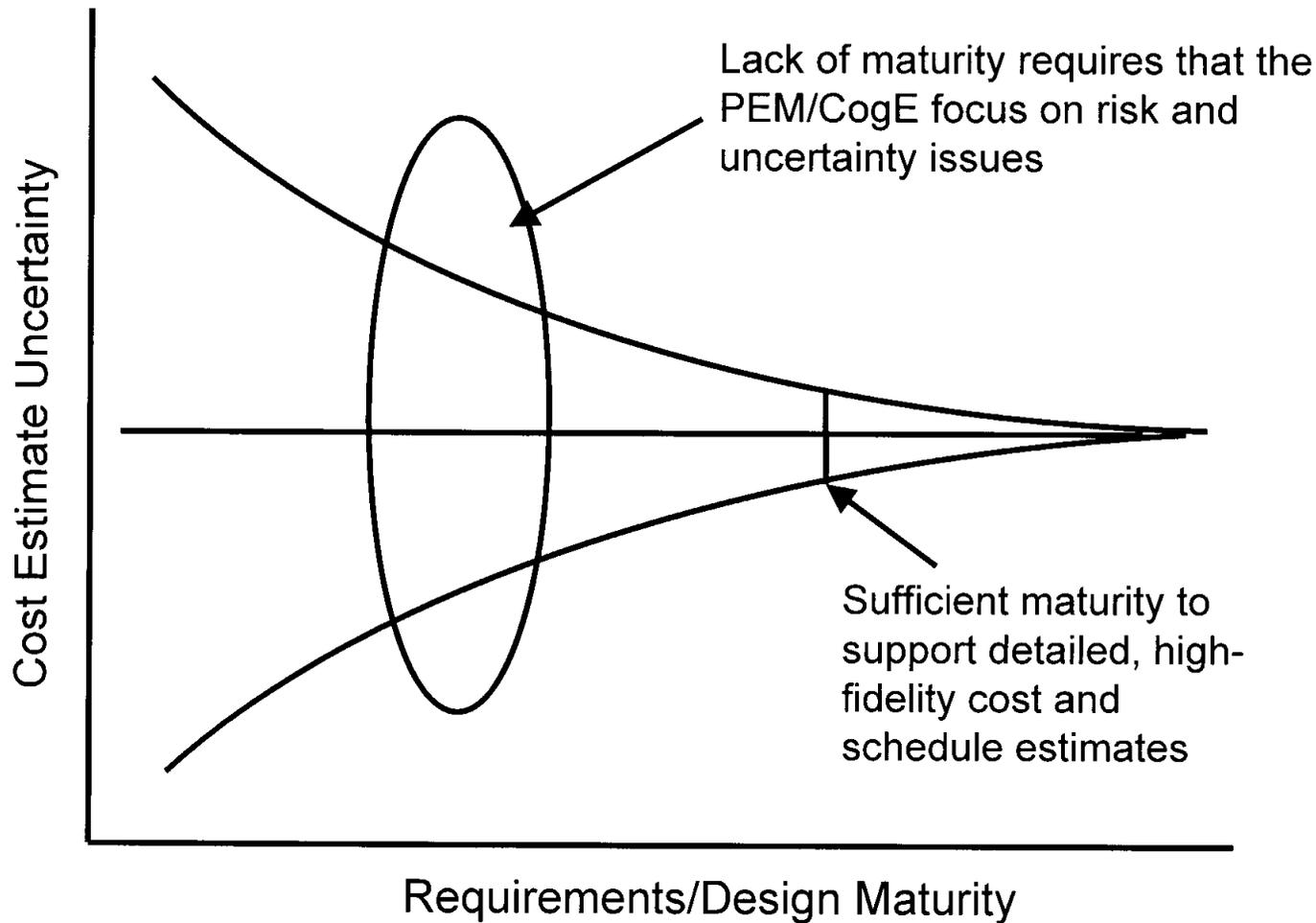
- It has been found that JPL software development projects overrun their planned effort as defined at PDR, excluding project/system-level reserves, by 50% on average
- The range covers everything from small under-runs of less than 10% to cost over-runs of well over 100% [Hihn and Habib-agahi, 2000]
- The various reasons for the systematic cost growth observed at JPL are also typical of software development throughout industry [Boehm, 2000]



# Cost Uncertainty: Accuracy in Estimating

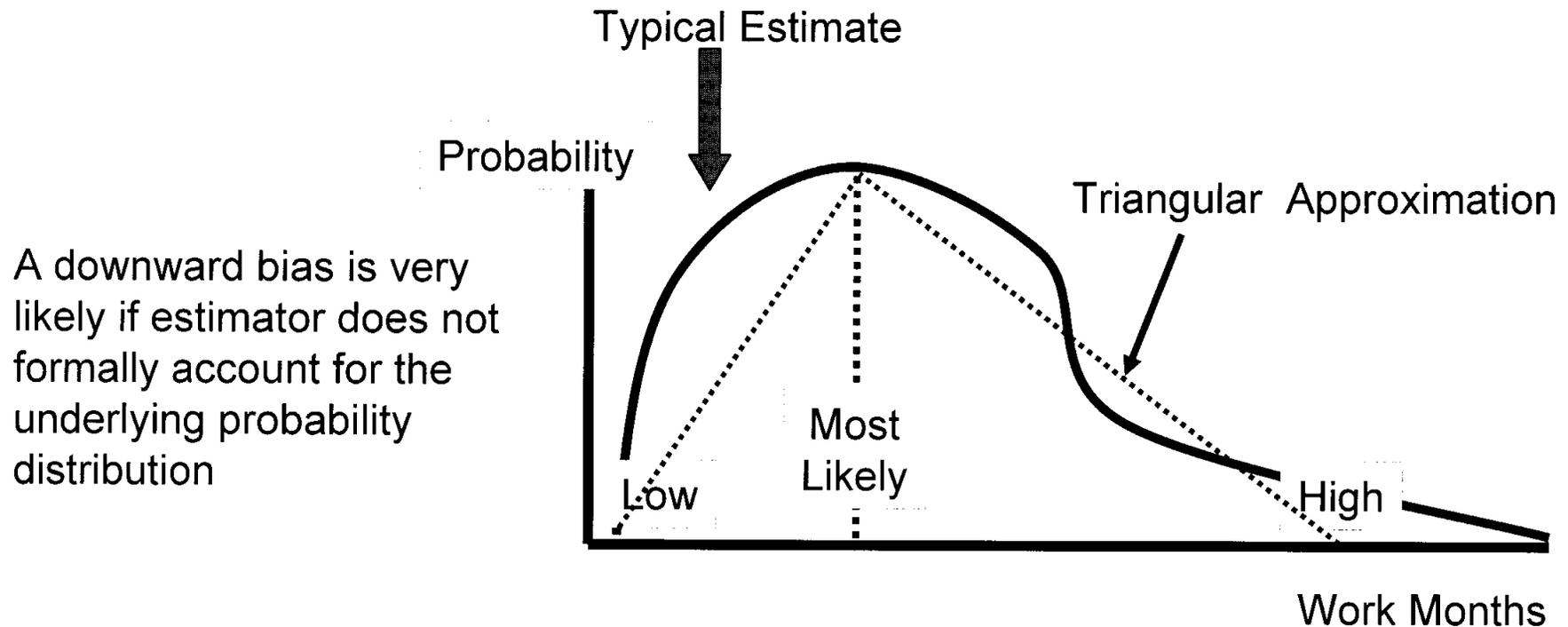


*Estimates Cannot be More Accurate than Requirements/Design Maturity*





# Cost Uncertainty: Fundamental Reason for Underestimation



- Typically cost, effort, SLOC distributions are highly skewed to the left
- We can capture all this with just three parameters
  - Low, Most Likely, and High
- Point estimates tend to fall between the low and most likely distribution parameters and Most Likely is typically less than 50<sup>th</sup> percentile



# Cost Uncertainty: Incorporating Uncertainty in Your Estimates



- There are several ways to address the under-estimation problem
  - One is to make all estimates as distributions and use Monte Carlo techniques to combine the estimated elements of the project
  - Another is to construct a risk adjusted Engineering Cost Estimate (ECE) derived from a quantitative risk list and your ECE
  - The second approach, which is simpler, is the standard Program Evaluation and Review Technique (PERT), a statistical method for estimating the mean of a triangular distribution:

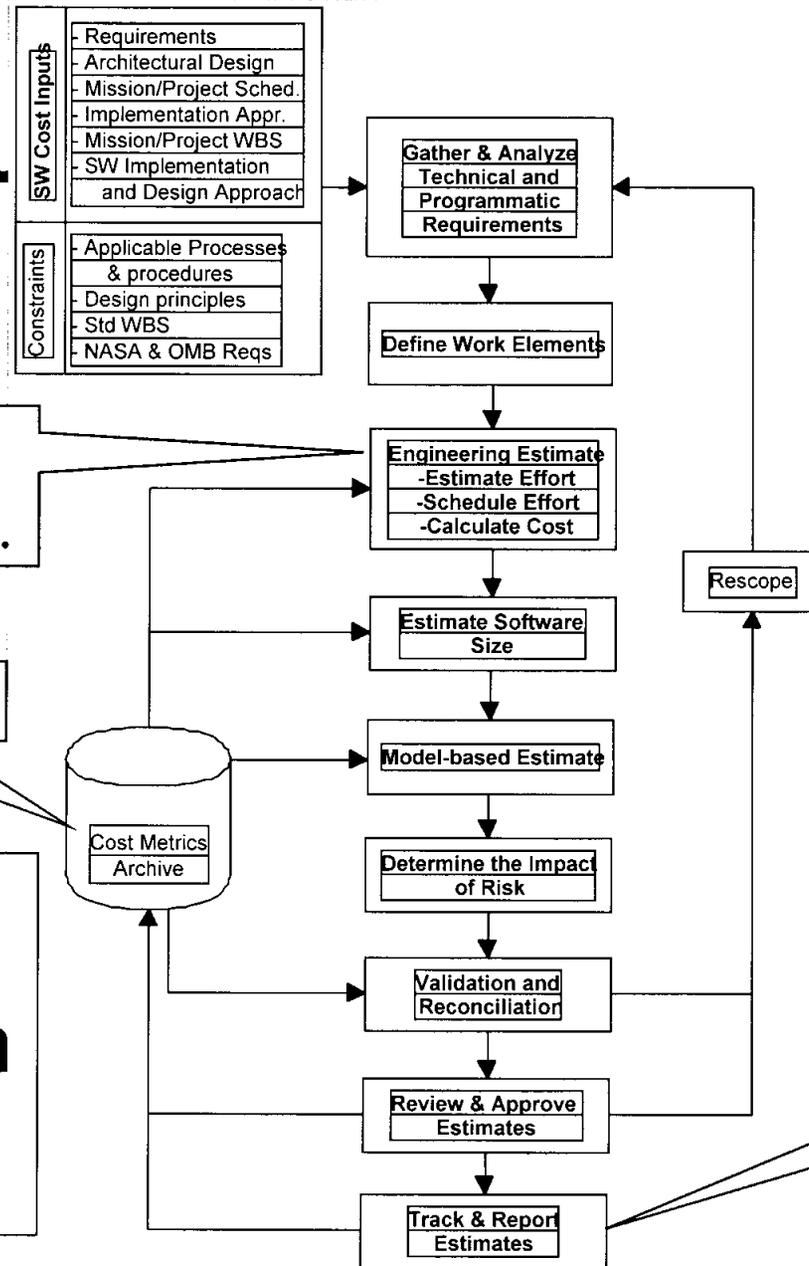
$$\text{Estimate} = \text{Mean} = (\text{Least} + 4 * \text{Most Likely} + \text{High}) / 6.$$



## Recommended Software Estimation Steps



- The cost estimation process is highly iterative
- The reason for the iteration over the different steps is that cost estimation is part of the larger planning and design process, in which the system is designed to fit performance, cost, and schedule constraints along with reconciliation and review of the different estimates
- The steps may be performed in a different order, the sequence described in the class is different from that shown in the handbook



When budget is too low  
 “Do not look for a silver bullet”  
 - DESCOPE

We do not cover this step in course.

Save History

# Software Estimation Steps

Follow Through



# What's Coming Next



- 
- The software estimation process discussed in the following lectures and exercises describes the steps for developing software estimates and subsequently tracking and refining those estimates throughout the life of the project