Scoping the Job

Presented by:
Jairus Hihn
JPL Software Quality Improvement Project

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

- Purpose
- Analyze and refine the requirements
  - Activities and products
- Define the Work Elements and Procurements
  - Activities and Products
- Begin identifying risk drivers
- Common mistakes in scoping
Software Estimation Steps
Define the Work Elements and Procurements

- The purpose of this step is to define the work elements and procurements for the software project that will be included in the software estimate.
Definition and Purpose

- Scoping is defined as this early planning activity where we are trying to determine the domain of the software task
  - What the task is, what it covers
  - But sometimes it is useful to address: "what it is not"
- Goal is to identify clearly and in sufficient details the entire set of software activities for the purpose of producing a good cost estimate
  - Producing a complete conceptual design is not the objective
  - It is important to determine the right level of task breakdown
- Scoping is NOT requirements writing. Requirements writing comes after scoping the product, developing the Ops Concept and defining the interfaces
- A more tangible product of scoping is the support for a workforce and lines of code estimate
- A less tangible product of scoping is to assist the estimation process in factoring for software complexity and productivity
Scoping all the “dimensions” of the software task

- The question of scope applies to all aspects of the software task, both products and activities:
  - The internal components of the task
  - The hardware environment
  - The test environment
  - The mission scenarios
  - The interfaces with other project elements
  - The computing resources and margins
  - The life-cycle of the software element. Also, the project life-cycle
  - The project risk and reserve postures

- Familiarity with the above software drivers is a job prerequisite for the cost estimator
A Good Software WBS

- Is a planning and reporting tool which outlines the list of work activities to complete the software job
  - Identifies all the activities required to do the software job
    - The sum of its elements is the entire software job
  - Top-down, with many levels of details
  - Outlined activities need to lend themselves to being costed

- Has the following properties:
  - Manageable – specific in authority and responsibility
  - Independent – few, but well defined interfaces with other elements
  - Tailorable – can be modified for unique system characteristics
  - Measurable – can be costed and progress can be tracked
Use a Product-Oriented Software WBS

- These work elements and procurements will typically fall into the following categories of a project-specific WBS:
  - Software Management
  - Software Systems Engineering
  - Software Functions
    - Fn1
    - Fn 2
    - etc
  - Software Development Test Beds
  - Software Integration and Test
  - Software System-level Test Support
  - Assembly, Test, & Launch Operations (ATLO) Support
  - Software Quality Assurance
  - IV&V
Other Software-related items

- Some of these software related cost items may be found elsewhere in the project WBS
  - Reserves
  - IV&V
  - Payload or instrument software
  - SQA
  - Device Driver software
  - Firmware
  - Training
Begin Identifying Cost Risk Drivers

- Identify the attributes of the work elements that will drive the size and effort estimates in terms of heritage and risk
- Examples are:
  - Anything that is new, such as code, language, or design method
  - High heritage elements
  - Possible reuse or use of COTS products
  - Criticality
  - Use of development tools
  - Concurrent development of hardware
  - Multiple implementing organizations
  - High complexity elements
  - Skill and experience level of team
Common mistakes in scoping

- Scope Creep
- Making bad or wrong assumptions
- Missing important features or cost drivers
- Focusing on features which are not important cost drivers