Preparing the project plan

Specifying tasks
Estimating tasks
Aggregating the estimates

The project plan

- We saw in session 1 that a project plan is . . .
- **a network of task specifications** with duration and cost estimates.

**What is a task specification?**

**What are two other common names for task network?**

**REVIEW Question**

- Q: What is a **task**?

- A: A unit of project work that:
  a. can be assigned to an individual
  b. is worth keeping track of

**Alternative names for a task network**

- **PERT** network (used mainly by military organizations)  **Project Evaluation and Review Technique**

- **WBS** (used mainly in the Project Management Institute)
  **Work Breakdown Structure**

Copyright 2003, Information Disciplines, Inc., Chicago
Defining a task
- A task specification defines precisely the work to be done.
- Specifically it contains:
  1. A brief description or title
  2. A list of the tangible results or deliverables to be produced
  3. A list of the prerequisite tasks that must be completed before this one can begin.
  4. Identification of the resources or skills required to perform it.
- How do we record that information
  - in a manual project plan?
  - in an automated PMS, such as *MS Project*?

How do we know when the prerequisite tasks are done?
- Easy: the task deliverables are available for use or inspection.
- How do we know when a task is 75% done?

Estimating a task
- We need good estimates of
  - The cost of performing the task
  - The minimum duration for performing the task
- Who should determine those estimates? When?
- Note that man-month or person-hours is not a reciprocal relationship.

Who makes the estimates and when?
- A project planning expert (often the project manager) makes the initial estimates in order to derive aggregate cost and target date.
- But when the task is assigned to a team member, that individual team member must agree to the commitment.
  *What if the team member and the project manager can't agree?*
No man-months

- If one senior programmer can design and develop a high-level application framework in 12 weeks,
  - How long will it take three programmers of comparable experience?
  - Why?

- What kinds of work can be done efficiently in parallel?

Aggregating the task estimates

- Is a bottom-up process
- You need the project plan first, i.e. the set of task specifications.

Aggregating cost and time for a project (or major phase of a project)

A. Cost and resources:
   - Just sum the products of resource quantities by resource costs

B. Duration:
   - Compute the critical path, the sum of the longest durations in the task network.
   - But that assumes unlimited resources!
   - The critical path represents the minimum time required. The actual time will depend on resource availability.

Estimating problem 1: Pressures on estimators

"The new inventory control system absolutely must be operational when we move into our new automated warehouse a year from September."
- edict from management

- Suppose the critical path yields completion the following January.
  - What should the project manager do?
  - What do typical project managers do?
Caving in to pressure
(Hoping for a miracle?)

- A year from September is a long way off. *Somehow* we'll meet the deadline.
- We can
  - work extra hard,
  - hire more programmers,
  - put in overtime,
  - etc.

*A crash project!*

---

Are crash projects ever justified?

- If there's a huge potential reward for success or a huge penalty for failure, we may charter a project to try on a **best effort** basis.
  - Provide top-quality support *such as?*
- If it then fails, the participants should be sincerely thanked for their efforts and in no case punished.
- If it succeeds the experience must not be taken as a new basis for reckless estimating of future projects.

---

Relationship to phase disciplines

- For a non-trivial system development project we never know enough at the beginning to prepare a detailed task network for the whole project.
- But the phase disciplines we looked at last time solve that problem. We estimate
  - the next phase in detail
  - subsequent phases roughly

*What did we call that strategy?*

---

Estimating problem 2: uncertain multipliers

- Certain tasks, especially in the early project phases, may spawn a variable number of other tasks, e.g.:
  - Task W: Identify next-level modules
  - Task Z1: Develop module A
  - Task Z2: Develop module B etc.
- How can we estimate those before we actually do task W?
- We may have to make good guesses based on experience *whose experience?*
Estimating problem 3: dependency on creativity

- A critical task, especially in a highly advanced application, may require coming up with:
  - some original idea,
  - a solution to a so-far unsolved problem,
  - research or experimentation with new technology,

*How can we estimate that?*