Software Engineering

SOFTWARE PROJECT PLANNING - II

RISK MANAGEMENT

PROJECT SCHEDULING

Gagan Deep
Founder & Director
Rozy Computech Services
Kurukshetra, rozygag@yahoo.com
www.rozyph.com

SOFTWARE RISK MANAGEMENT

What is Risk?

- Risk concerns future happenings. Tomorrow's problems are today's risk.
- Hence, a simple definition of a "risk" is a problem that could cause some loss or threaten the success of the project, but with has not happened yet.
- These potential problems might have an adverse impact on cost, schedule, or technical success of the project, the quality of our software products, or project team morale.
- Risk management is the process of identifying addressing and eliminating these problems before they can damage the project.

- We need to differentiate risks, as potential problems, from the current problems of the project.
- Different approaches are required to address these two kinds of issues.
- For example, a staff shortage because we have not been able to hire people with the right technical skills is a current problem; but the threat of our technical people being hired away by the competition is a risk.
- Current real problems require prompt, corrective action, whereas risk can be dealt with in several different ways.
- We might choose to avoid the risk entirely by changing the project approach or even canceling the project.

Typical Software Risks

- The list of evil things that can befall a software project is depressingly long.
- Possible risks can come from group brainstorming activities, or from a risk factor chart accumulated from previous projects.
- There are no magic solutions to any of these risk factors, so we need to rely on past experience and a strong knowledge of contemporary software engineering and management practices to control these risks.

Capers Jones has identified the top five risk factors that threaten projects in different applications .

- Dependencies
- Requirement Issues
- Management Issues
- Lack of Knowledge
- Other Risk Categories
  - Unavailability of adequate testing facilities
  - Turnover of essential personnel
  - Unachievable performance requirements
  - Technical approaches that may not work

Preventive measure to reduce Risks : Broadly, there are five strategies for risk reduction:

- Hazard prevention
- Likelihood reduction
- Risk avoidance
- Risk transfer
- Contingency planning

PROJECT SCHEDULING

First, let us discuss the meaning of scheduling.

- Scheduling is the proper distribution of time and effort.
- It also describe, what activity is to be performed at which time e.g. schedule of classes in the school/college.
- Similar is the meaning of schedule in software project planning. In software project schedule, the estimated time (schedule) is divided according to the phases of development.

- Scheduling for software engineering projects can be viewed from two rather different perspectives.
- In the **first view**, an end-data for release of a computer-based system has already (and irrevocably) been established. The software organization is constrained to distribute effort within the prescribed time frame.
- The **second view** of software scheduling assumes that rough chronological bounds have been discussed but that the end-data is set by the software engineering organization. Effort is distributed to make best use of resources and an end-data is defined after careful analysis of the software.
- Unfortunately, the first situation is encountered far more frequently than the second.

Basic Principles of Software Project Scheduling

- ***Compartmentalization :*** The project must be compartmentalized into a number of manageable activities and tasks. To accomplish compartmentalization, both the product and the process are decomposed.
- ***Interdependency :*** The interdependency of each compartmentalized activity or tasks must be determined. Some tasks must occur in sequence while others can occur in parallel.
- ***Time allocation: E****ach task must be assigned a start date and a completion date that are a function of the interdependencies.

- ***Effort validation:*** As time allocation occurs, the project manager must ensure that no more than the allocated number of people have been scheduled at any given time.
- ***Defined responsibilities:*** Every task that is scheduled should be assigned to a specific team member.
- ***Defined outcomes:*** Every task that is scheduled should have a defined outcome. For software projects, the outcome is normally a work product (e.g., the design of a module) or a part of work. Work products are often combined in deliverables.
- ***Defined milestones:*** Every task or group of tasks should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

TOOLS & TECHNIQUE OF PROJECT SCHEDULING

There are mainly two techniques :

- PERT & CPM Chart and
- Timeline Chart

PERT & CPM Chart

- *Program Evaluation and Review Technique* (PERT) and *Critical Path Method* (CPM) are two project methods that can be applied to software development.

Both techniques are driven by information already developed in earlier project planning activities:

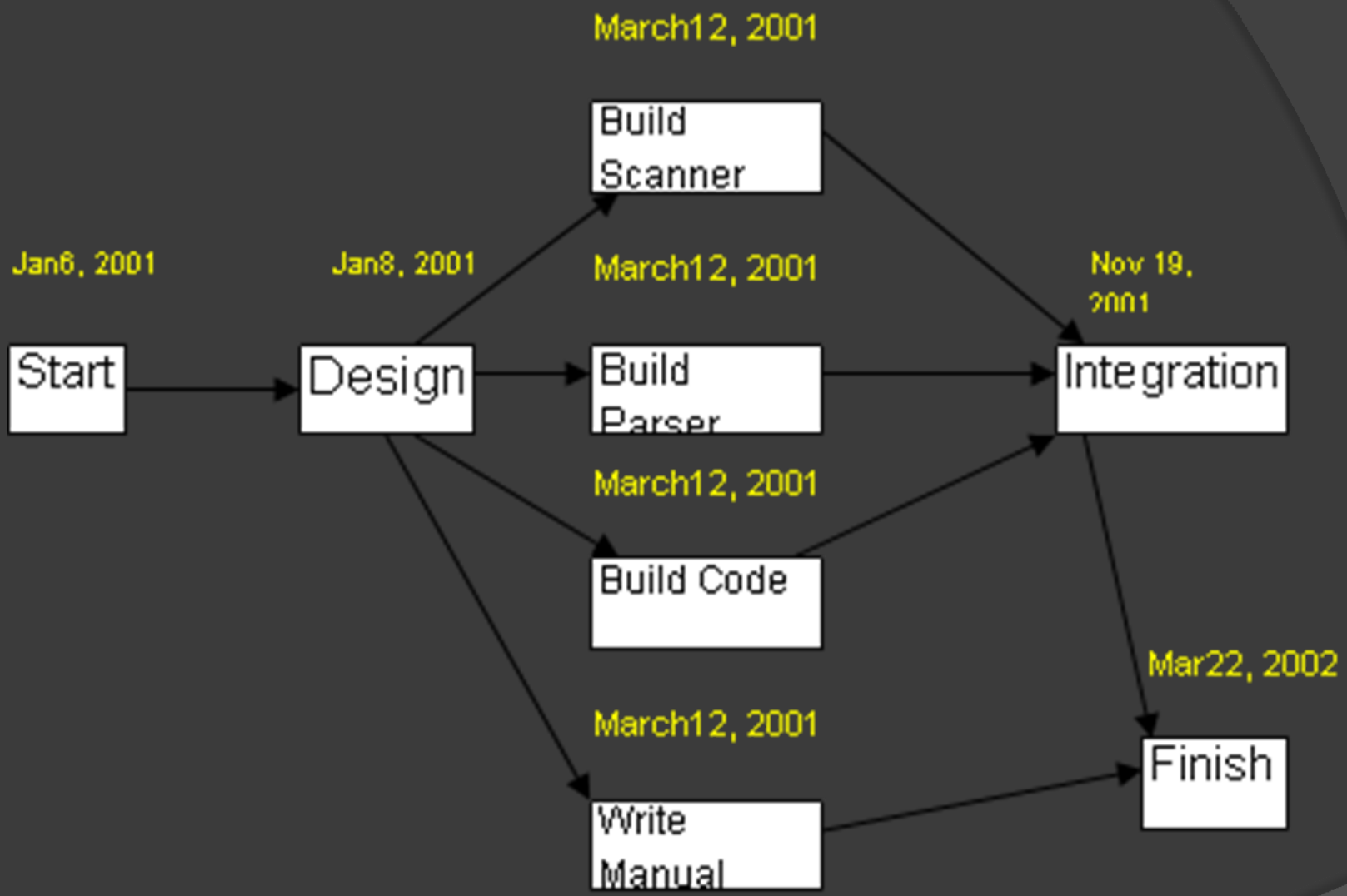
- Estimates of effort
- A decomposition of the production function
- The selection of the appropriate process model and task set
- Decomposition of tasks.

Both PERT and CPM provide quantitative tools that allow the software planner to

- **Determine** the *critical path* - the chain of tasks that determines the duration of the project;
- **Establish** "most likely" time estimates for individual tasks by applying statistical models; and
- **Calculate** "boundary times" that define a time "window" for a particular task.

Example of PERT

- A PERT chart is a network of boxes (or circles) and arrows.
- There are different variations of PERT charts.
- Some use the boxes top represent activities, and some use the arrows to do so. We will use the first approach here.
- Each box thus represents an activity.
- The arrows are used to show the dependencies of activities on one another.
- The activity at the head of an arrow cannot start until the activity at the tail of the arrow is finished.
- Some boxes can be designated as milestones.



PERT chart for a simple compiler project.

Advantages of PERT

- It forces the manager to plan.
- It shows the interrelationships among the tasks in the project and, in particular, clearly identifies the critical path of the project, thus helping to focus on it.
- It exposes all possible parallelism in the activities and thus helps in allocating resources.
- It allows scheduling and simulation of alternative schedules.
- It enables the manager to monitor and control the project.

Timeline Charts

- When creating a software project schedule, the planner begins with a set of tasks (the work breakdown structure
- As a consequence of this input, a *timeline chart,* also called a *Gantt chart*, is generated.
- A timeline chart can be developed for the entire project.

- GANTT charts can be used for project planning showing project activities and time required. The steps to be performed are:
- Identify which tasks are to be performed by which date.
- Identify tasks which can be performed at the same time.
- Identify tasks which are dependent on completion of some other tasks.
- Plot activities on GANTT Chart.

Thanks!

rozygag@yahoo.com
www.rozyph.com