Lecture- 10

Project Scheduling

Dronacharya College of Engineering

Scheduling and Planning

- The majority of projects are 'completed' late, if at all.
- A project schedule is required to ensure that required project commitments are met.
- A schedule is required to track progress toward achieving these commitments.

Why Software Is Delivered Late?

- ✓ An unrealistic deadline
- ✓ Changing but unpredicted customer requirements
- ✓ Underestimation of efforts needed
- ✓ Risks not considered at the project start
- ✓ Unforeseen technical difficulties
- ✓ Unforeseen human difficulties
- ✓ Miscommunication among project staff
- ✓ Failure to recognize that project is falling behind schedule

"One day at a time"

- **All technical projects involve 100s of small tasks**
- **Some tasks do no affect the project completion**
- **Other tasks are *critical* for project completion**
- **Project manager must:**
  - define all project tasks
  - build a network that depicts their interdependence
  - identify the *critical* tasks
  - track the progress of these tasks
  - recognize the delay *"one day at a time"*

Two different Perspectives to view Scheduling

End date for completion has been finalized

Only Rough time-frame is given

People and Effort

"If we fall behind schedule we can always add more programmers and catch to late in the project"

Has a disruptive effect on the project

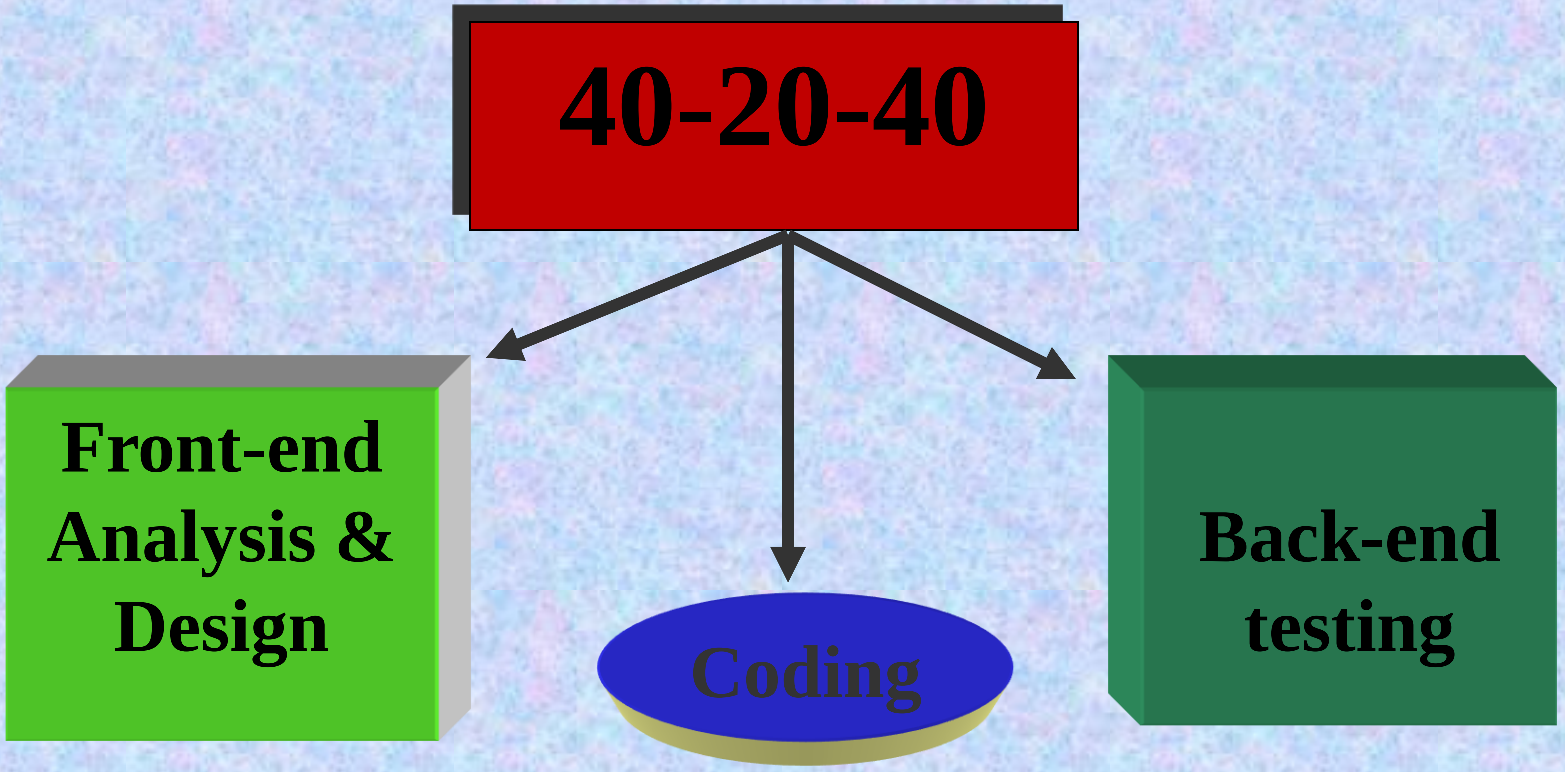
Schedules slip even further

People and Effort

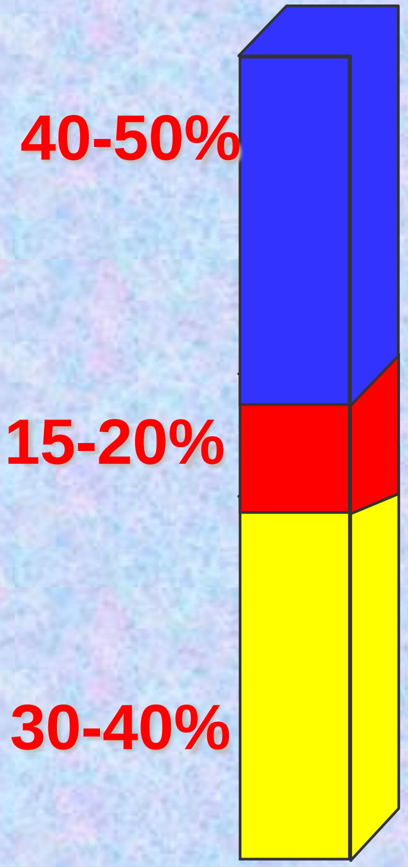
The relationship between the number of people working in software project and overall productivity is
not linear

Fewer people and ***longer time period*** is a better option for software development

Effort Distribution



Effort Allocation



- **“front end” activities**
  - customer communication
  - analysis
  - design
  - review and modification
- **construction activities**
  - coding or code generation
- **testing and installation**
  - unit, integration
  - white-box, black box
  - regression

Basic Principles for SE Scheduling

- Compartmentalization – define distinct tasks
- Interdependency- parallel and sequential tasks
- Time allocation - assigned person days, start time, ending time)
- Effort validation - be sure resources are available
- Defined responsibilities — people must be assigned
- Defined Outcomes- each task must have an output
- Defined milestones - review for quality

Scheduling and Planning

In order to make a schedule, the following tasks must be completed:

- Identify manageable activities and tasks by decomposing the process and the product.
- Determine which tasks are dependent on the completion of others. (Which activities must occur in sequence and which can occur concurrently.)
- Allocate each task a number of work-units (often person-days), a start date and a completion date.
- Define responsibilities for the tasks (allocate them to a person or persons).
- Define outcomes of the tasks (deliverables) and milestones for the schedule.
- Review the proposed tasks, their effort allocation and start and end dates with the people involved to ensure there are no conflicts and over allocation.

Identifying Tasks

The first step :

Identify The Tasks Required To Be Performed.

These tasks will comprise software engineering activities broken down for product functions.

A schedule is not a fixed entity and as such it will be refined as a project progresses.

- Initially rough

  a project schedule usually refers to the work tasks, deliverables and milestones for major software engineering activities and major product functions
- is refined in detail

  as the project progresses to refer to specific tasks and activities that must be completed for those major activities and functions.

Selecting Project Tasks

No Set Of Tasks Is Appropriate For All Projects.

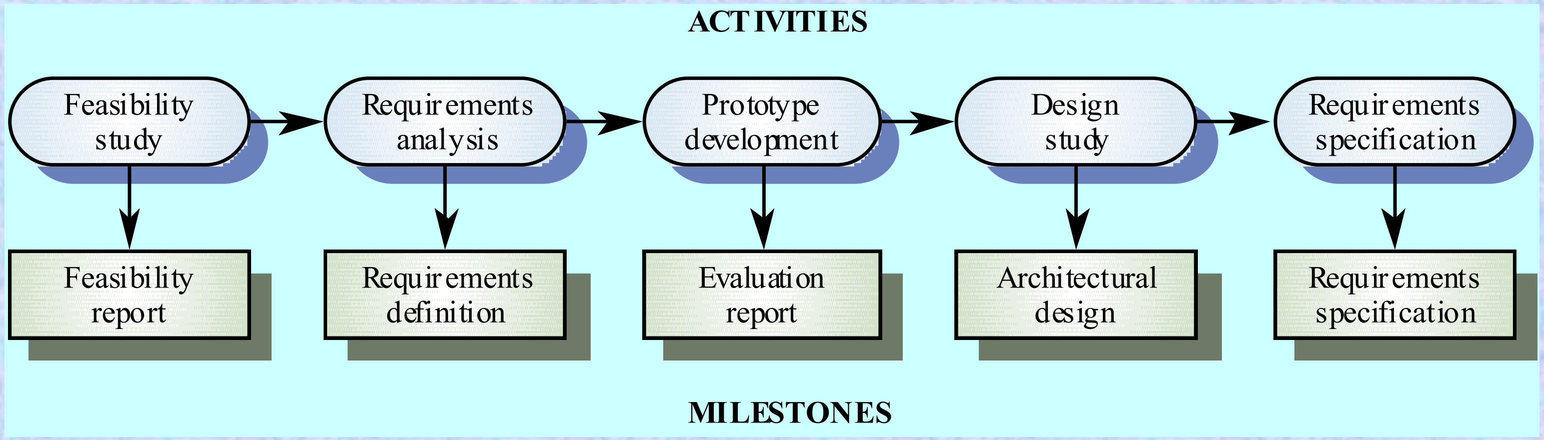
The set of tasks that are appropriate for a project depends on a number of factors. These include:

- **The process model selected**. An iterative development model would require different tasks, etc... than would a waterfall model or rapid application development model.
- **The type of project**. A new development project has a different set of tasks to a maintenance project or to a concept development.
- **The size and complexity** of the product.
- **The rigor required in development**. This is a factor generally determined by things like product size, mission criticality, stability of requirements, etc...

Selecting Project Tasks

- ***Milestone*** = end-point of a specific, distinct software process activity or task (for each milestone a report should be presented to the management)
- ***Deliverable*** = project result delivered to the client

In order to establish milestones the phases of the software process phases need be divided in basic activities/tasks.



Defining Task Sets

- determine type of project
- assess the degree of rigor required
  - identify adaptation criteria
  - compute task set selector (TSS) value
  - interpret TSS to determine degree of rigor
- select appropriate software engineering tasks

Software Project Types

- Concept Development projects
- New Application Development Projects
- Application Enhancement Project
- Application Maintenance Project
- Reengineering Project

Degree of Rigor

Casual

• Minim
required

•Umbrel
minimiz

•Docume
requiren
reduced

•Basic pr
SE are a

Structure

•Framewor
will be appl

•Umbrella t
applied

•Documenta
measureme
will be done
streamlines

Strict

•Full proc
applied

•Degree o
ensures h

•All umbr
will be ap

•Robust v
will be pr

Quick Reaction

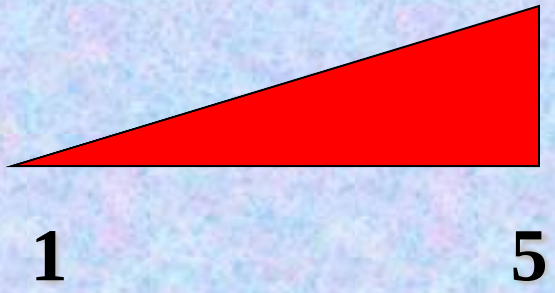
•Process framework will be applied

•Only essential tasks will be undertaken

• Documentation will be provided after product delivery

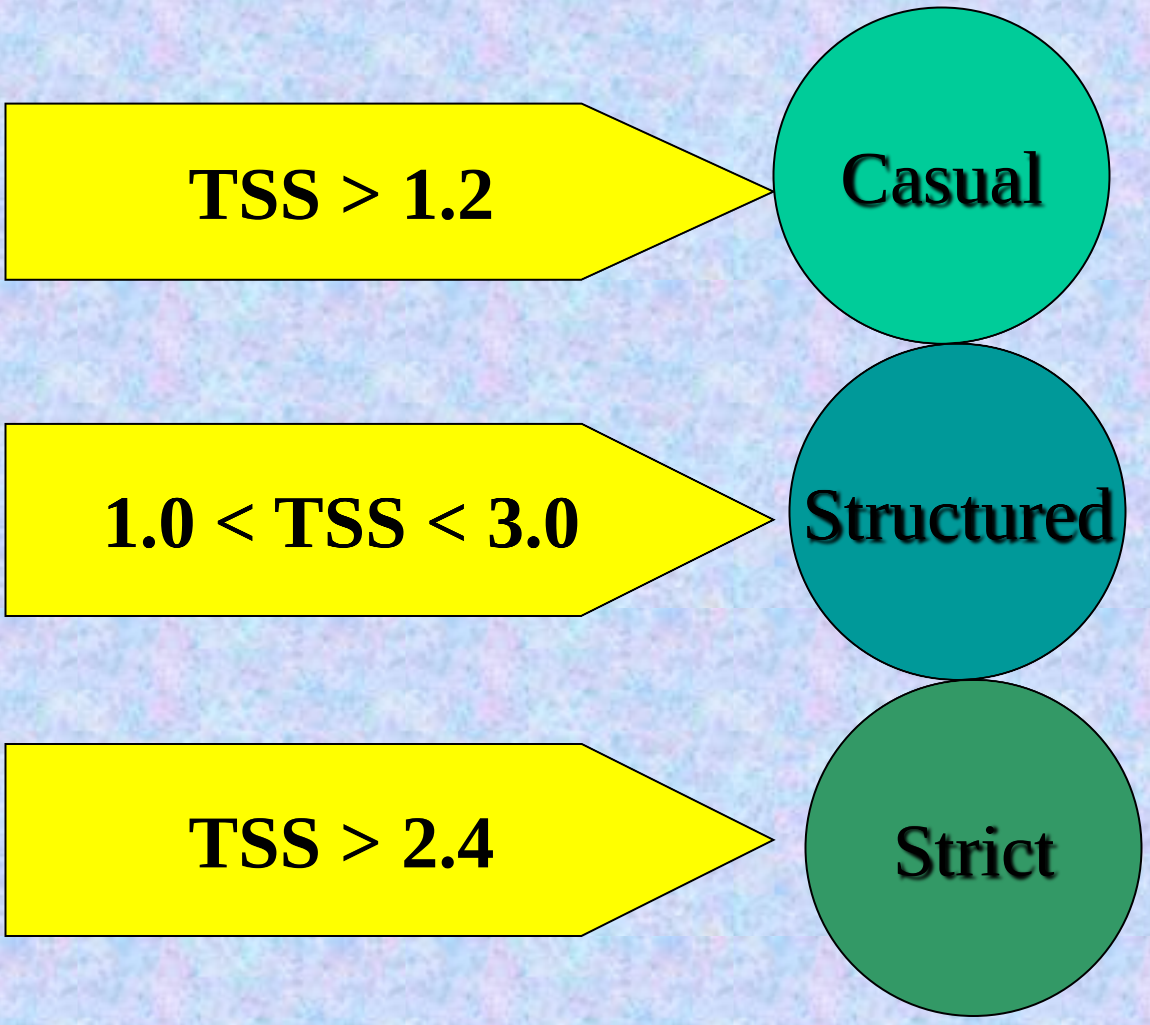
Adaptation Criteria

- ✓ Size of the Project
- ✓ Number of potential users
- ✓ Mission criticality
- ✓ Application Longevity
- ✓ Stability requirements
- ✓ Ease of communication
- ✓ Maturity of technology
- ✓ Performance constraints
- ✓ Embedded and non-embedded characteristics
- ✓ Project staff
- ✓ Reengineering factors



Task Set selector

- Based on adaptation criteria, TSS is computed



Activity Network Diagrams

An activity network diagram provides a notation for documenting

- a network of tasks needed to complete a project,
- their interdependencies
- the times required for each task.

There are a number of activity network techniques which are similar in nature. The most commonly used include PERT (Project Evaluation and Review Technique) and CPM (Critical Path Method).

Pert Chart

A simple PERT chart comprises circles (nodes) to represent events within the development lifecycle

For example commencement / completion of tasks, and lines (edges) which represent the the tasks. The lines are additionally labeled by the estimated duration of the task.

Note: there are a number of variations to this notation. A real PERT chart shows earliest time to completion, latest time to completion, and slack in the circles also.

How to construct a PERT chart

The basic steps to constructing a PERT chart are:

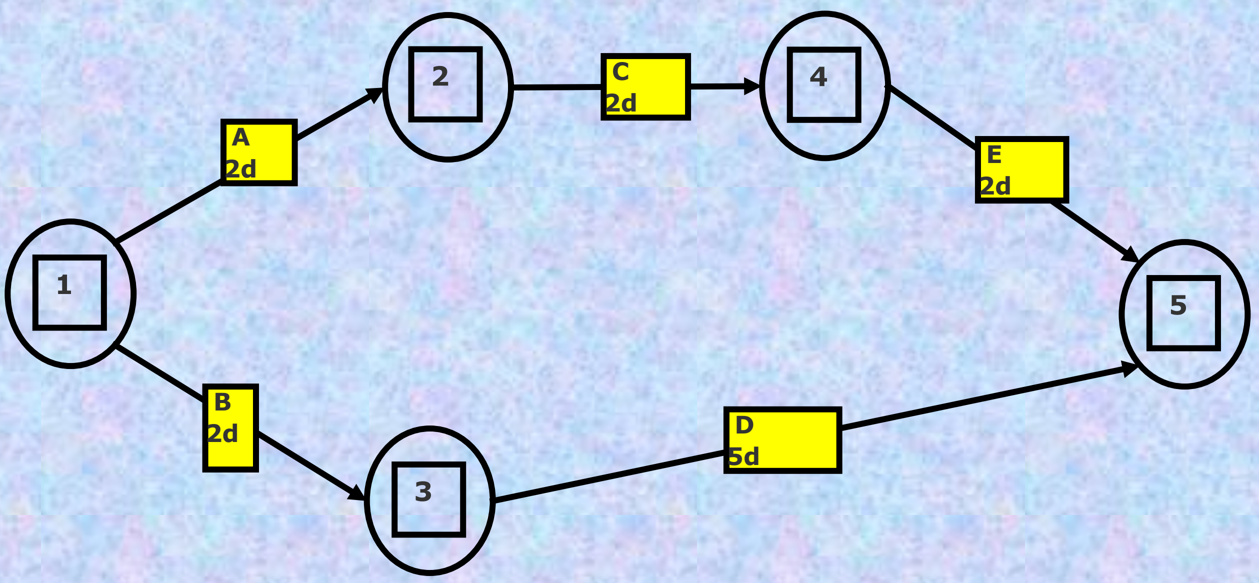
- Identify tasks and estimate duration of times
- Identify a single start and end event
- Arrange events in sequence (give events a unique number)
- Establish start and finish times of each task. Keep in mind the estimates made for duration and effort.
- Determine float
- Revise

Pert Chart

As an example of using a PERT chart, consider the following simple chart showing a project with **tasks A,B,C,D and E**

This diagram states that **tasks A,B,C and E will take 2 days** (assume d is abbreviation for days) and **task D has a planned duration of 5 days.**

Task D is dependent on completion of task B, etc.



The Critical Path

The critical path is the path between the start event and end event which takes the longest time.

Note that:

- No task on the critical path can take longer without extending the end date of the project.
- Tasks on the critical path are called critical tasks.
- No critical task can have any slack.
- Tasks on the critical path must be carefully monitored.

The Critical Path

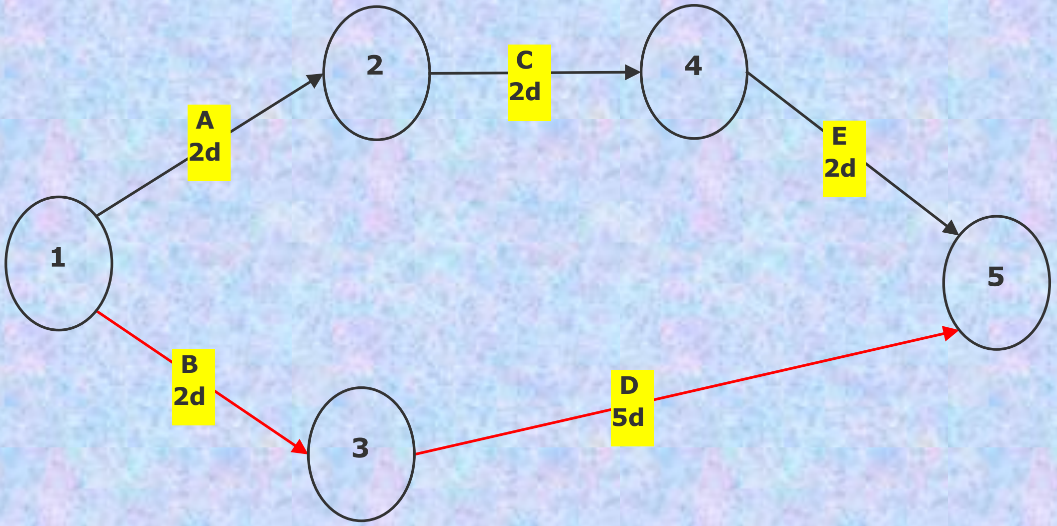
In the example above the critical path can be described by events 1,3 and 5 or by tasks B,D.

This is because the time to reach the end event (5) on this path is longer than any other path. This means that task B must take no longer than 2 days and task D no longer than 5 days or the end date for event E will need to be extended.

The duration of the other path is 6 days. Because the critical path is 7 days, there is slack (or float) of one day on the other path.

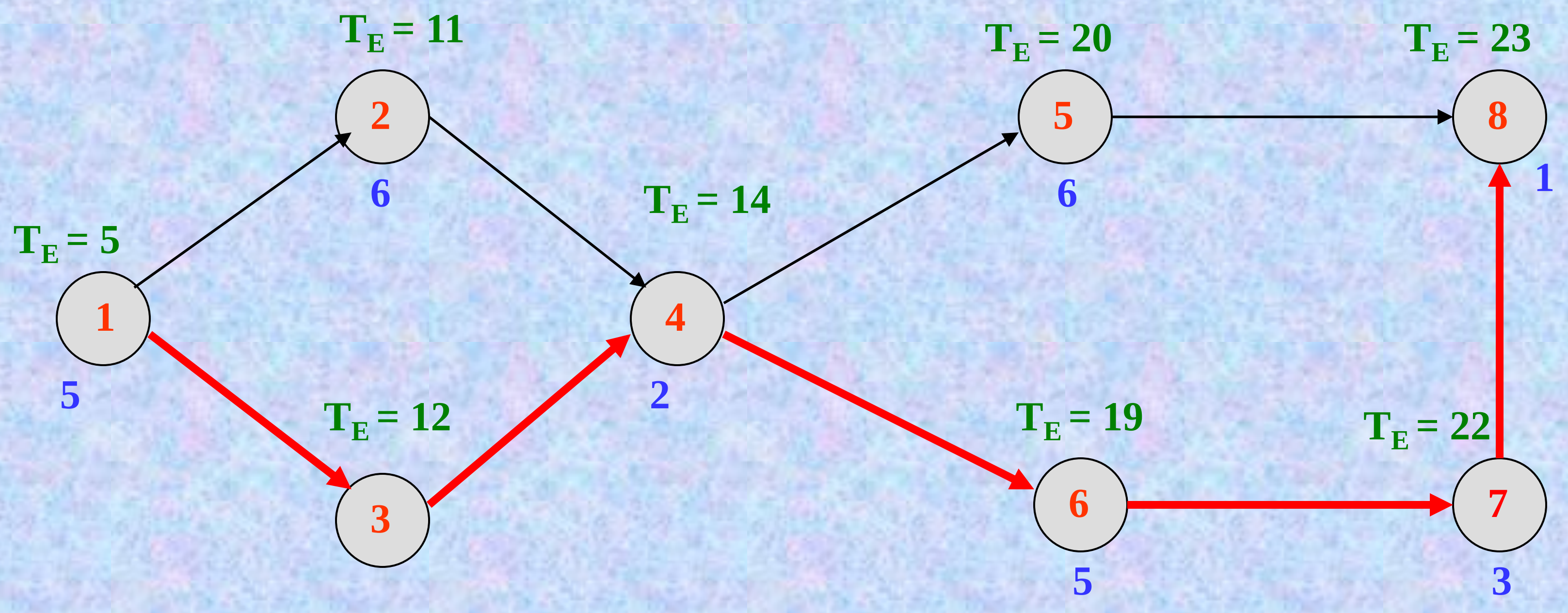
This means that this path can take 1 day longer than planned.

That is, any one task on this path (A,C or E) can take 1 day longer than expected. Note this slack must be shared between the tasks on this other path. They can not all take an extra day



Example:

PERT/CPM Chart



Task. A project has been defined to contain the following list of activities along with their required times for completion:

Activity No	Activity	Expected completion time	Dependency
1.	Requirements collection	5	-
2.	Screen design	6	1
3.	Report design	7	1
4.	Database design	2	2,3
5.	User documentation	6	4
6.	Programming	5	4
7.	Testing	3	6
8.	Installation	1	5,7

a. Draw a PERT chart for the activities.

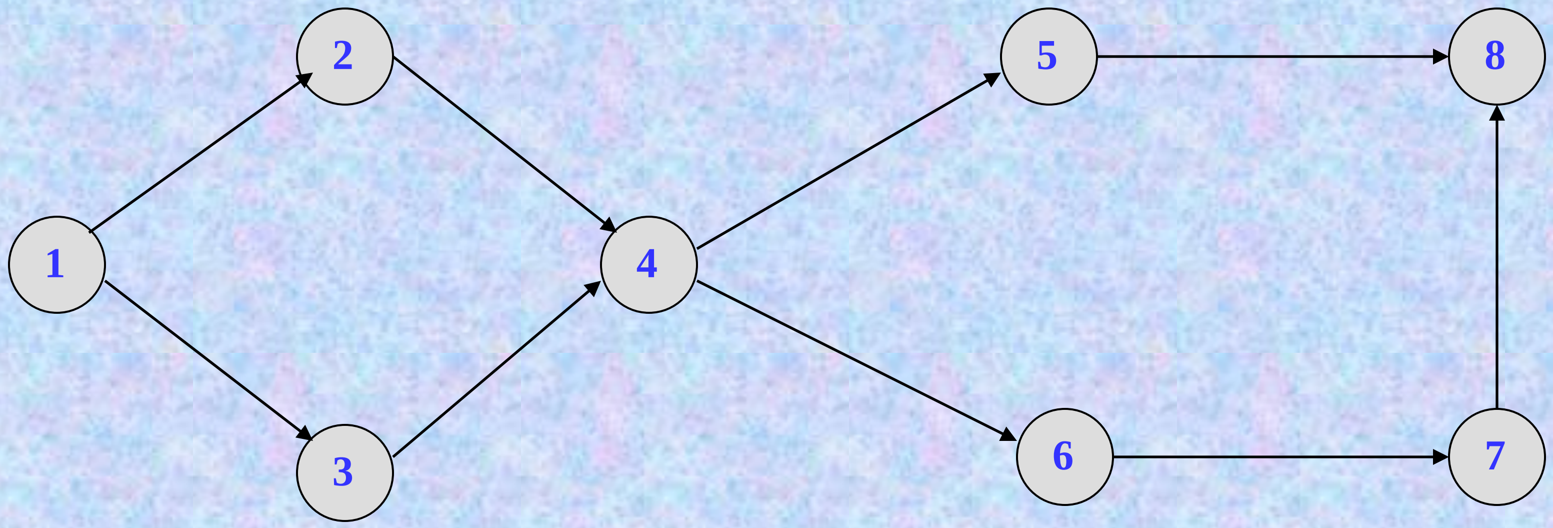
b. Calculate the earliest expected completion time.

c. Show the critical path.

PERT/CPM Chart (cont'd)

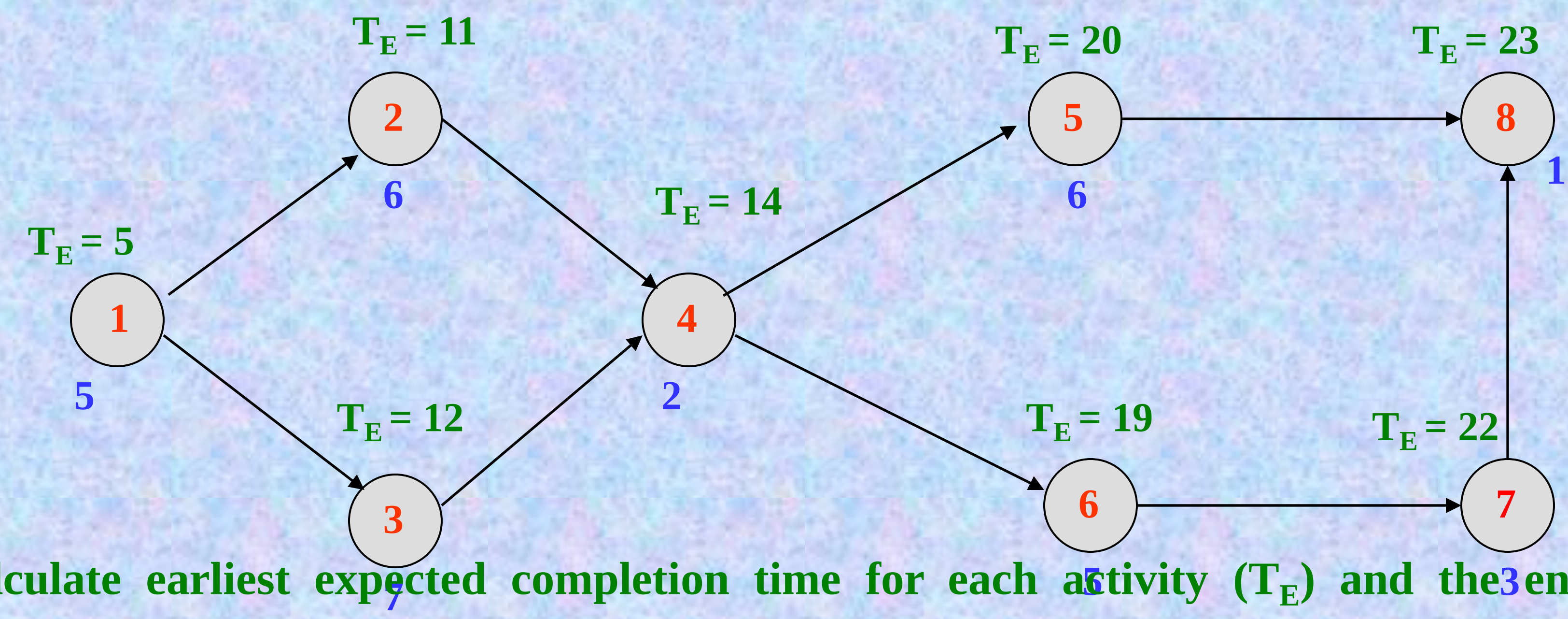
a. Draw a PERT chart for the activities.

Using information from the table, show the sequence of activities.



b. Calculate the earliest expected completion time.

1. Using information from the table, indicate expected completion time for each activity.



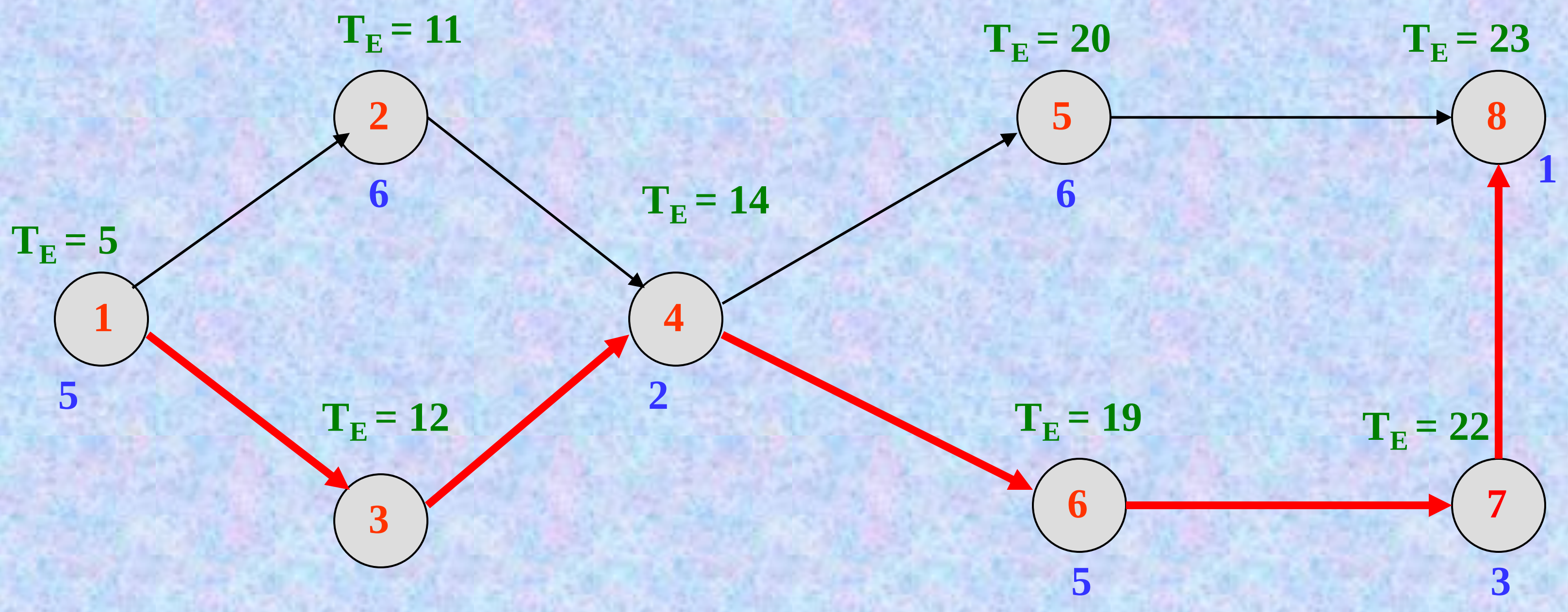
2. Calculate earliest expected completion time for each activity (T_E) and the entire project.

Hint: the earliest expected completion time for a given activity is determined by summing the expected completion time of this activity and the earliest expected completion time of the immediate predecessor.

Rule: if two or more activities precede an activity, the one with the largest T_E is used in calculation (e.g., for activity 4, we will use T_E of activity 3 but not 2 since 12 > 11).

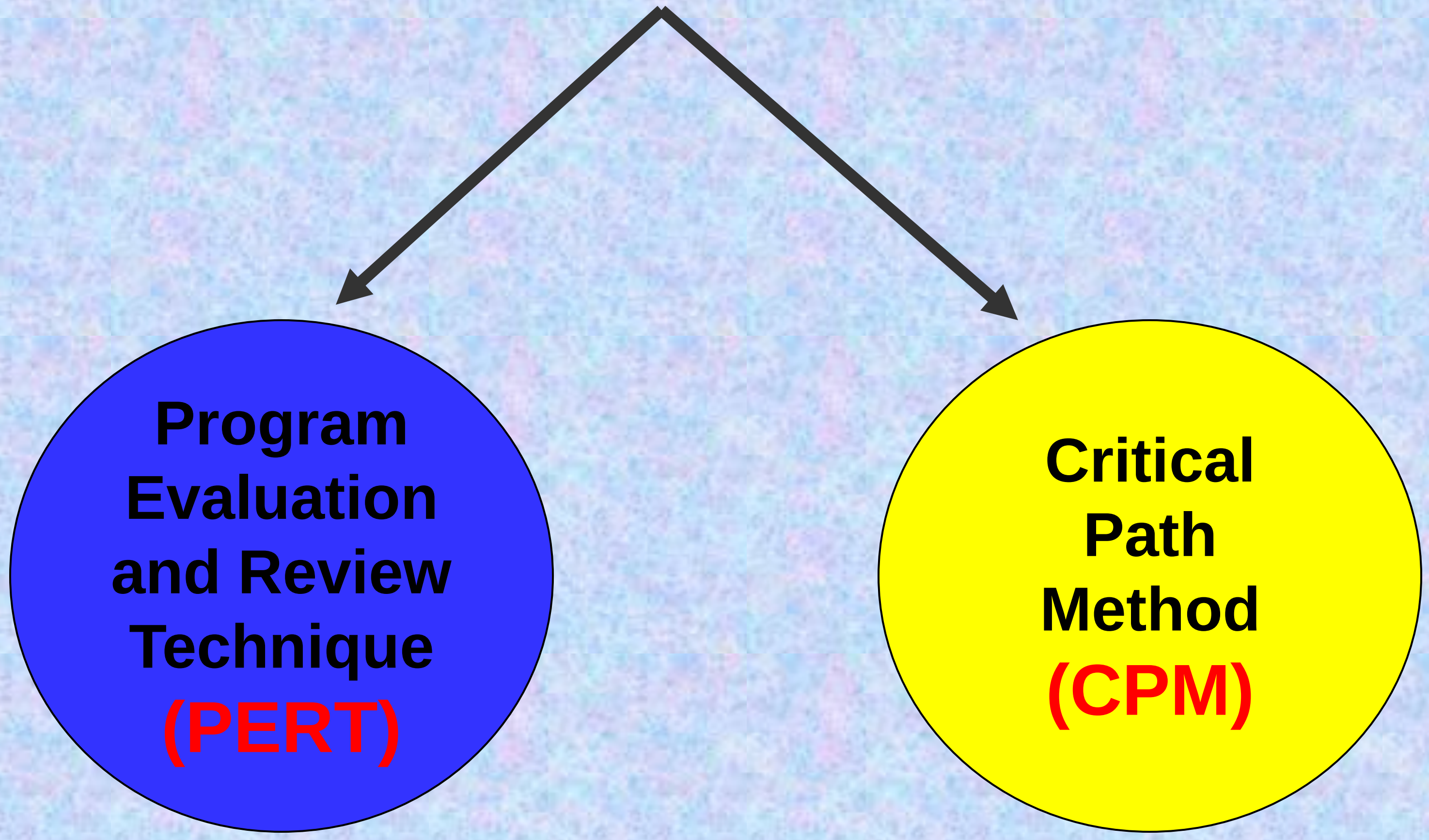
PERT/CPM Chart (the end)

c. Show the critical path.



The critical path represents the shortest time, in which a project can be completed. Any activity on the critical path that is delayed in completion, delays the entire project. Activities not on the critical path contain slack time and allow the project manager some flexibility in scheduling.

Project Scheduling


Program Evaluation and Review Technique (PERT)
Critical Path Method (CPM)