

# **Project Scheduling & Tracking (Week 3)**

# Planning, Estimating, Scheduling

---

- ▶ What's the difference?
- ▶ Estimating: Determining the size & duration of activities.
- ▶ Plan: Identify activities. No specific start and end dates.
- ▶ Schedule: Adds specific start and end dates, relationships, and resources.



# How To Schedule

---

- ▶ 1. Identify “what” needs to be done
  - ▶ Work Breakdown Structure (WBS)
- ▶ 2. Identify “how much” (the size)
  - ▶ Size estimation techniques
- ▶ 3. Identify the dependency between tasks
  - ▶ Dependency graph, network diagram
- ▶ 4. Estimate total duration of the work to be done
  - ▶ The actual schedule



# Partitioning Your Project

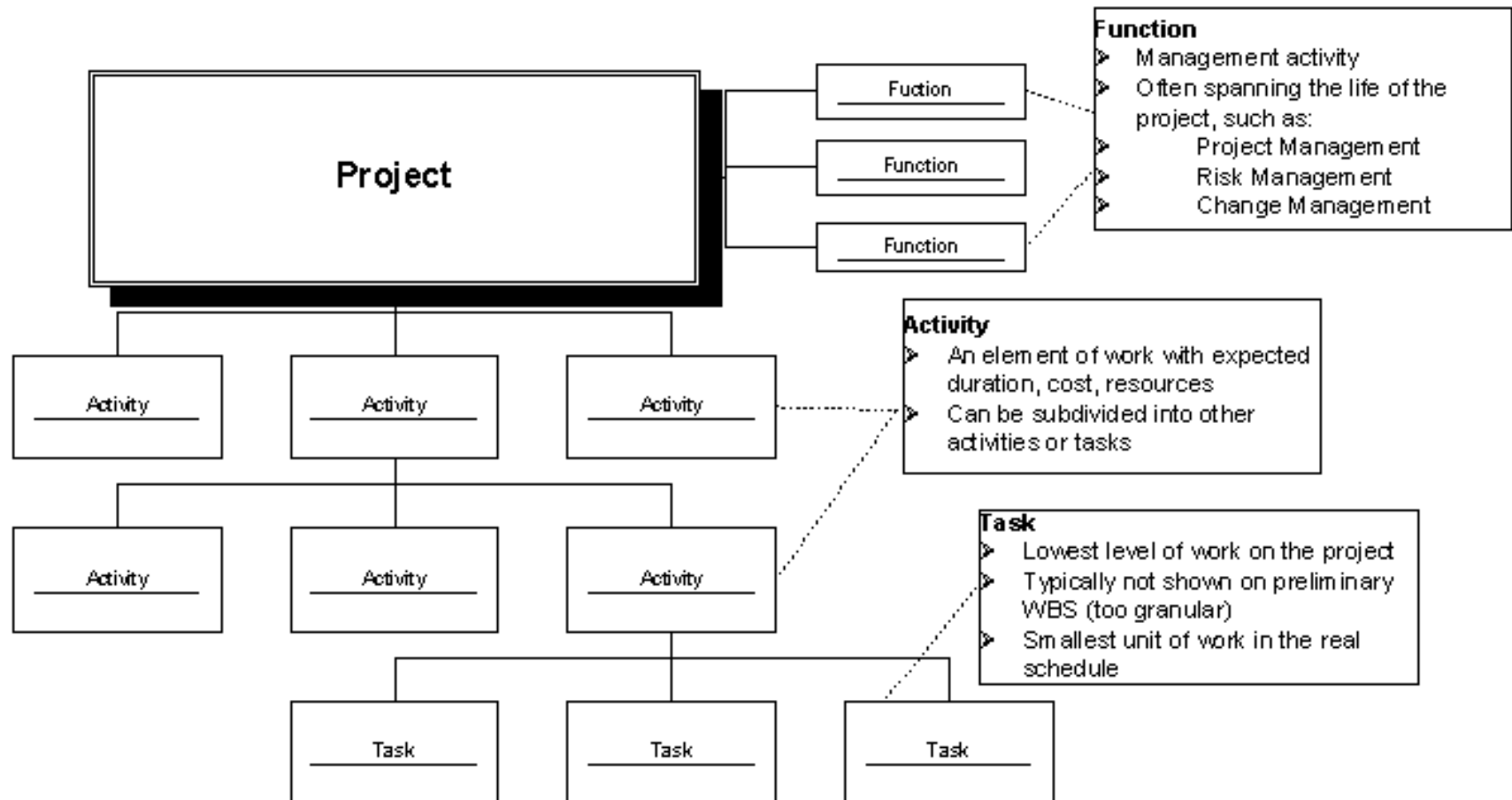
---

- ▶ You need to decompose your project into manageable chunks
- ▶ ALL projects need this step
- ▶ Divide & Conquer
- ▶ Two main causes of project failure
  - ▶ Forgetting something critical
  - ▶ Ballpark estimates become targets
- ▶ How does partitioning help this?



# Project Elements

## ► A Project: functions, activities, tasks



# Work Break Down Structure (WBS)

---

- ***Work Break Down Structure*** – a check list of the work that must be accomplished to meet the project objectives.
- The WBS lists the major project outputs and those departments or individuals primarily responsible for their completion.



# WBS Outline Example

---

0.0 Retail Web Site

1.0 Project Management

2.0 Requirements Gathering

3.0 Analysis & Design

4.0 Site Software Development

4.1 HTML Design and Creation

4.2 Backend Software

4.2.1 Database Implementation

4.2.2 Middleware Development

4.2.3 Security Subsystems

4.2.4 Catalog Engine

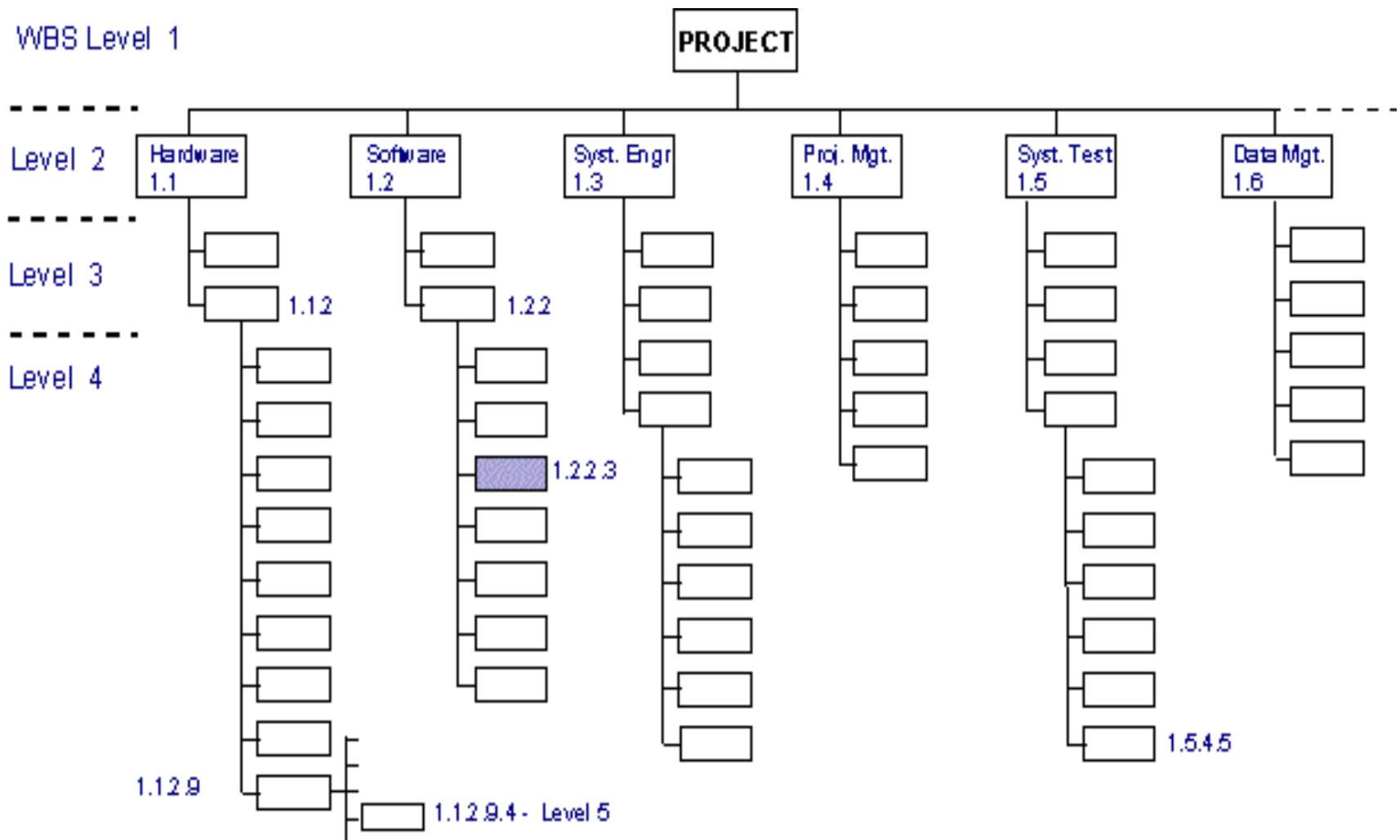
4.2.5 Transaction Processing

4.3 Graphics and Interface

4.4 Content Creation

5.0 Testing and Production





From: [http://www.hyperthot.com/pm\\_wbs.htm](http://www.hyperthot.com/pm_wbs.htm)



# WBS Types

---

## ▶ Process WBS

- ▶ a.k.a Activity-oriented
- ▶ Ex: Requirements, Analysis, Design, Testing
- ▶ Typically used by PM

## ▶ Product WBS

- ▶ a.k.a. Entity-oriented
- ▶ Ex: Financial engine, Interface system, DB
- ▶ Typically used by engineering manager

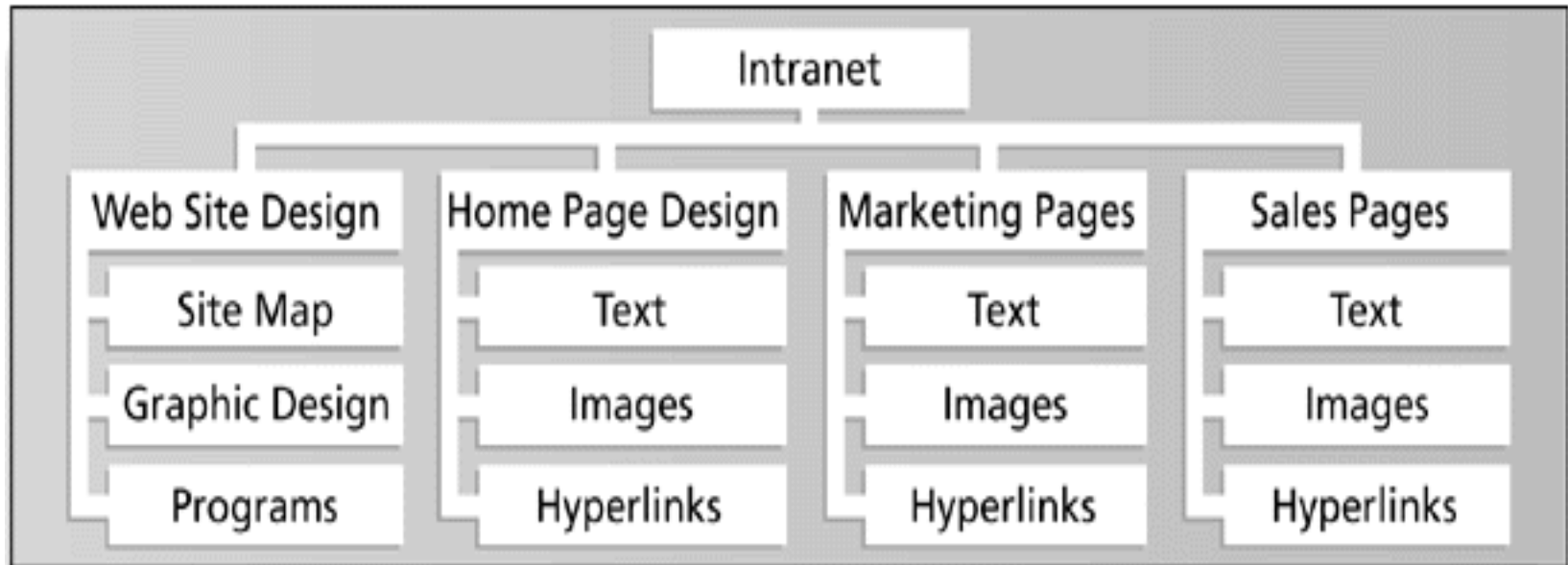
## ▶ Hybrid WBS: both above

- ▶ This is not unusual
- ▶ Ex: Lifecycle phases at high level with component or feature-specifics within phases
- ▶ Rationale: processes produce products



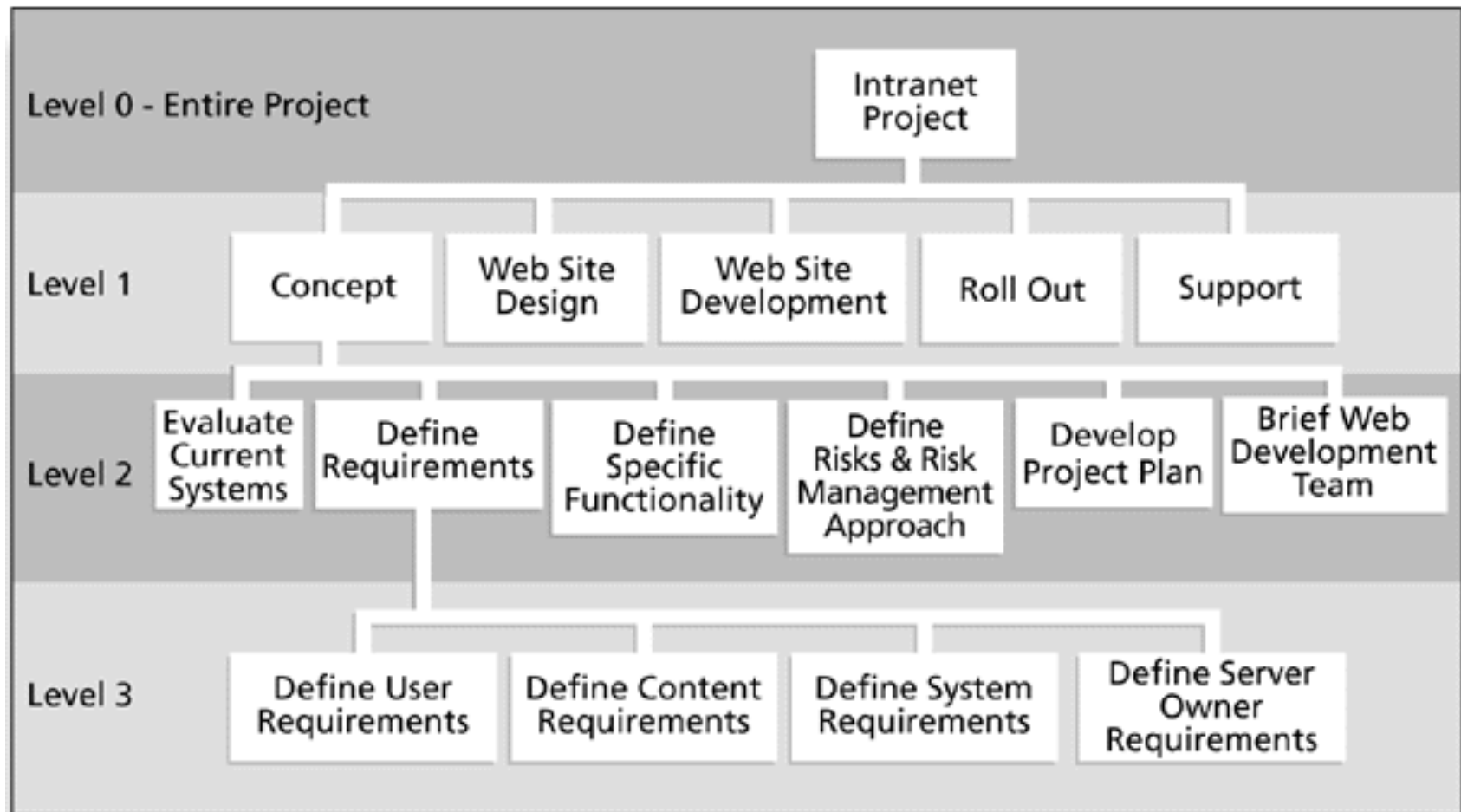
# Product WBS

---



# Process WBS

---



# WBS

---

- ▶ List of Activities, not Things
- ▶ List of items can come from many sources
  - ▶ SOW, Proposal, brainstorming, stakeholders, team
- ▶ Describe activities using “bullet language”
  - ▶ Meaningful but terse labels
- ▶ All WBS paths do not have to go to the same level
- ▶ Do not plan more detail than you can manage



# Work Packages (Tasks)

---

- ▶ Generic term for discrete **tasks** with definable end results
- ▶ The “one-to-two” rule
  - ▶ Often at: 1 or 2 persons for 1 or 2 weeks
- ▶ Basis for monitoring and reporting progress
  - ▶ Can be tied to budget items (charge numbers)
  - ▶ Resources (personnel) assigned
- ▶ Ideally shorter rather than longer
  - ▶ Not so small as to micro-manage



# WBS Techniques

---

- ▶ Top-Down
- ▶ Bottom-Up
- ▶ Analogy
- ▶ Rolling Wave
  - ▶ 1<sup>st</sup> pass: go 1-3 levels deep
  - ▶ Gather more requirements or data
  - ▶ Add more detail later
- ▶ Post-its on a wall



# WBS Techniques

---

## ▶ Analogy

- ▶ Base WBS upon that of a “similar” project
- ▶ Use a template
- ▶ Analogy also can be estimation basis
- ▶ Pros
  - ▶ Based on past actual experience
- ▶ Cons
  - ▶ Needs comparable project



# Sequence the Work Activities

---

- ✦ Milestone Chart
- ✦ Gantt chart
- ✦ Network Techniques
  - CPM (Critical Path Method)
  - PERT (Program Evaluation and Review Technique)





# Gantt Chart

---

- Gantt chart is a means of displaying simple activities or events plotted against time or dollars
- Most commonly used for exhibiting program progress or for defining specific work required to reach an objective
- Gantt charts may include listing of activities, activity duration, scheduled dates, and progress-to-date





# Gantt Chart

---

- Advantages:
  - Easy to understand
  - Easy to change
- Disadvantages:
  - only a vague description of the project
  - does not show interdependency of activities
  - cannot show results of an early or late start of an activity

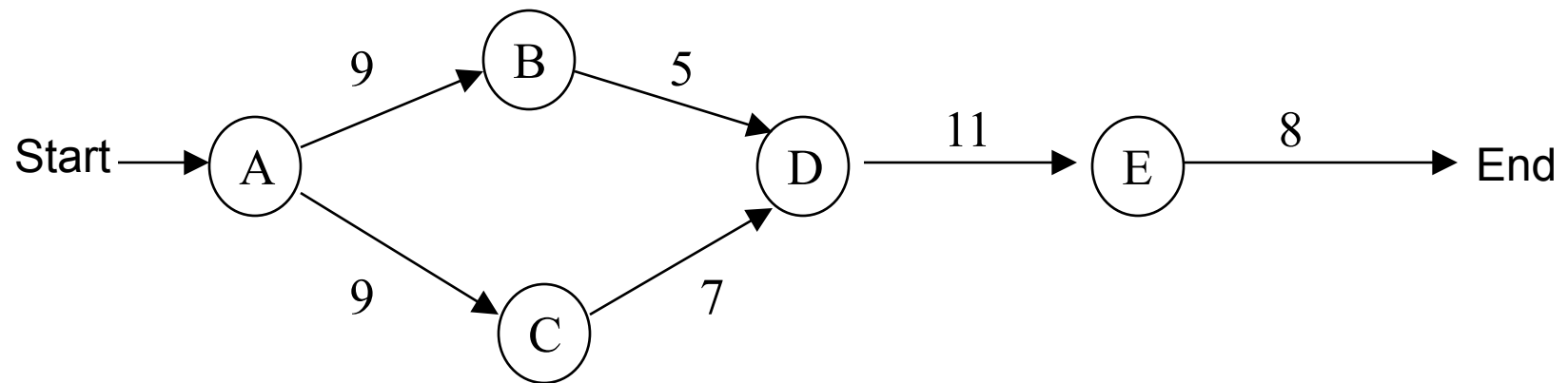
# Network Techniques

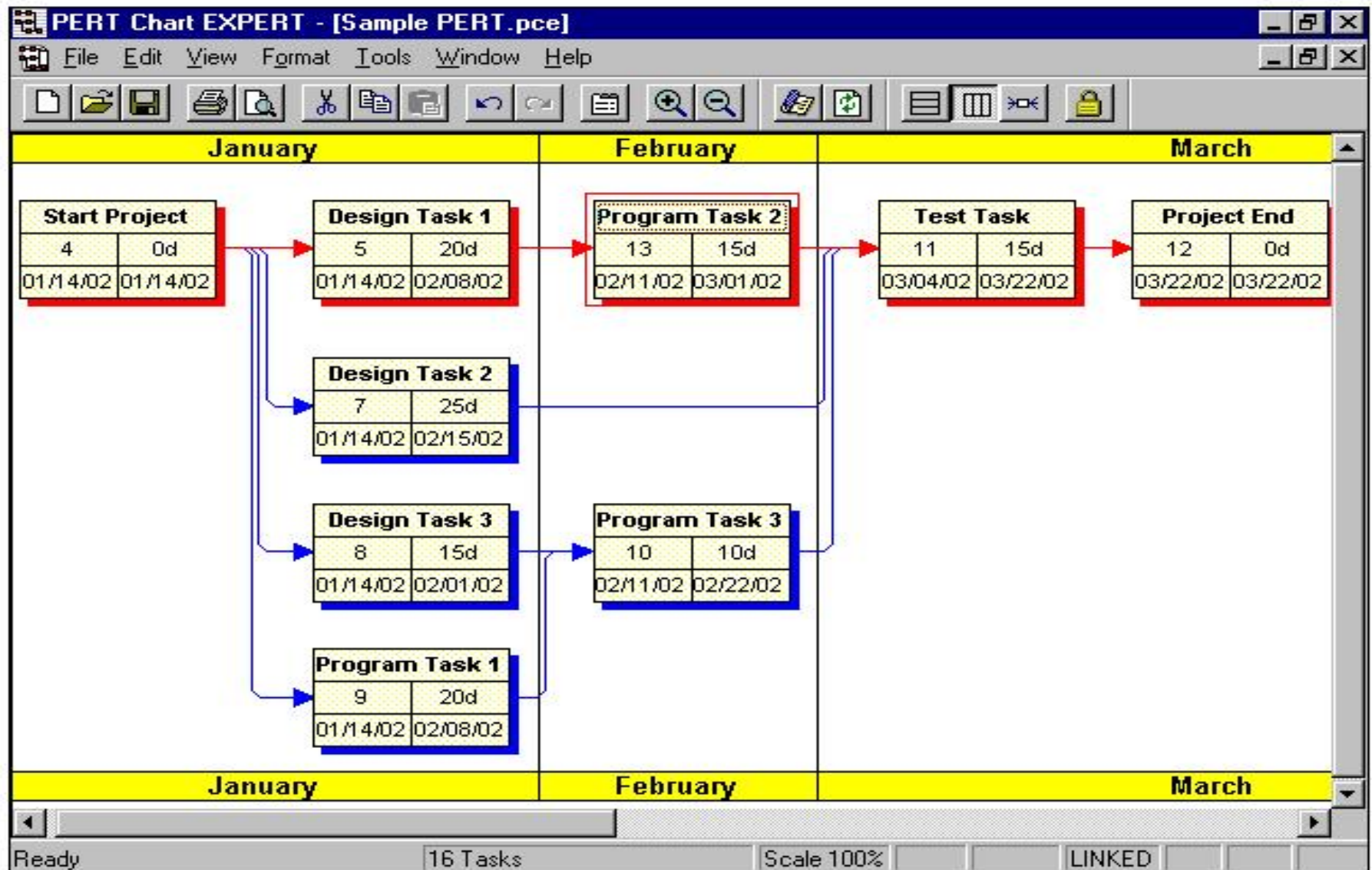
---

- A *precedence network* diagram is a graphic model portraying the sequential relationship between key events in a project.
- Initial development of the network requires that the project be defined and thought out.
- The network diagram clearly and precisely communicates the plan of action to the project team and the client.



Task	Duration	Dependencies
A - Architecture & design strategy	9	start
B - Decide on number of releases	5	A
C - Develop acceptance test plan	7	A
D - Develop customer support plan	11	B,C
E - Final sizing & costing	8	D





# CPM

---

Critical Path Method (CPM) tries to answer the following questions:

1. What is the duration of the project?
2. By how much (if at all) will the project be delayed if any one of the activities takes  $N$  days longer?
3. How long can certain activities be postponed without increasing the total project duration?



# Critical Path

---

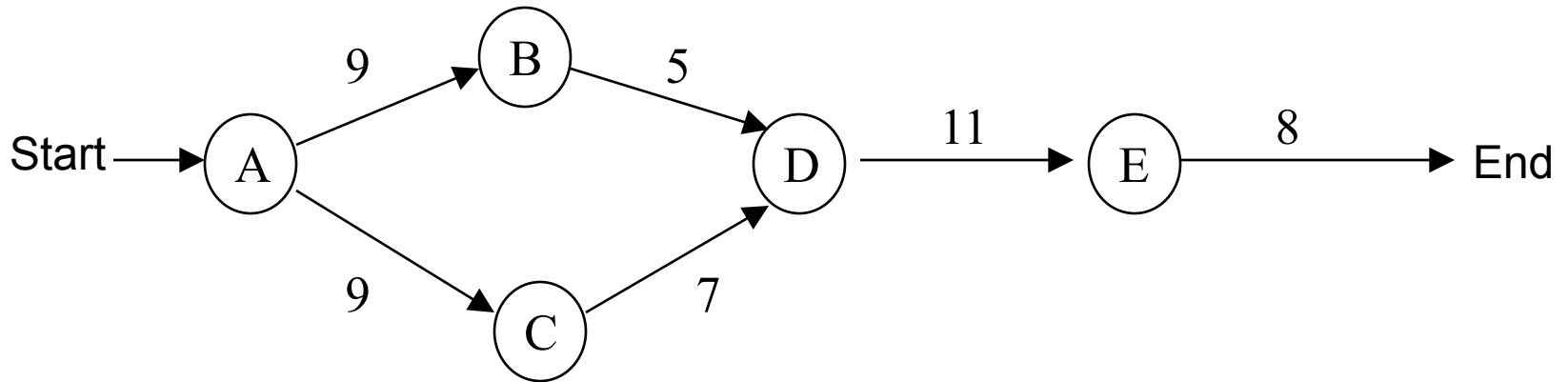
- Sequence of activities that have to be executed one after another
- Duration times of these activities will determine the overall project time, because there is no slack/float time for these activities
- If any of the activities on the critical path takes longer than projected, the entire project will be delayed by that same amount
- Critical path = Longest path in the precedence network (generally, the longest in time)





# Critical Path

---



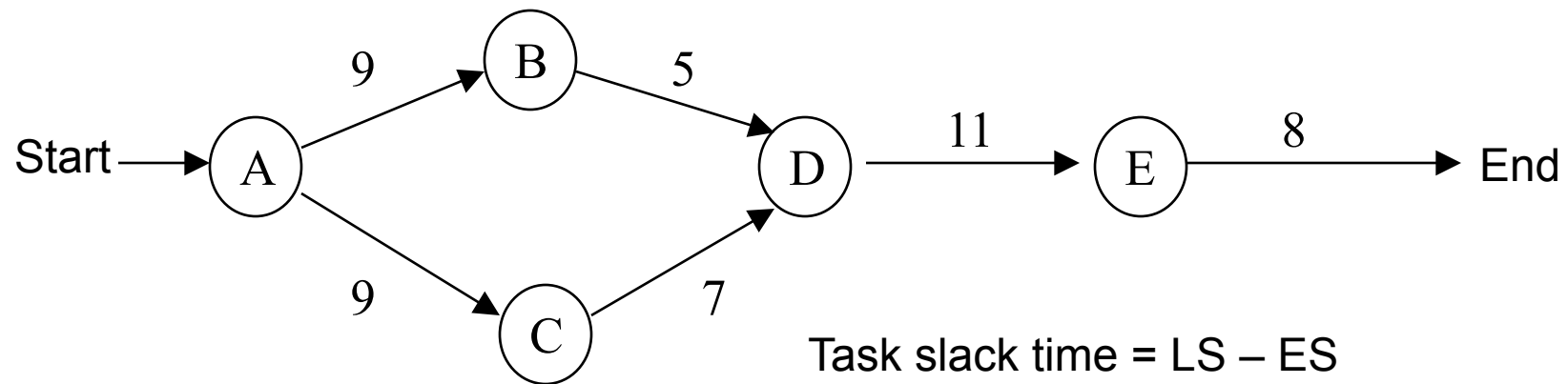
Critical Path = A – C – D – E (35 time units)

Critical Tasks = A,C,D,E

Non-Critical Path = A-B-D-E



Task	Duration	Depend	Earliest Start	Earliest Finish	Latest Start	Latest Finish
A	9	none	0	9	0	9
B	5	A	9	14	11	16
C	7	A	9	16	9	16
D	11	B,C	16	27	16	27
E	8	D	27	35	27	35



**Slack time** – maximum allowable delay for a non-critical activity.

$$\text{Task slack time} = \text{LS} - \text{ES}$$

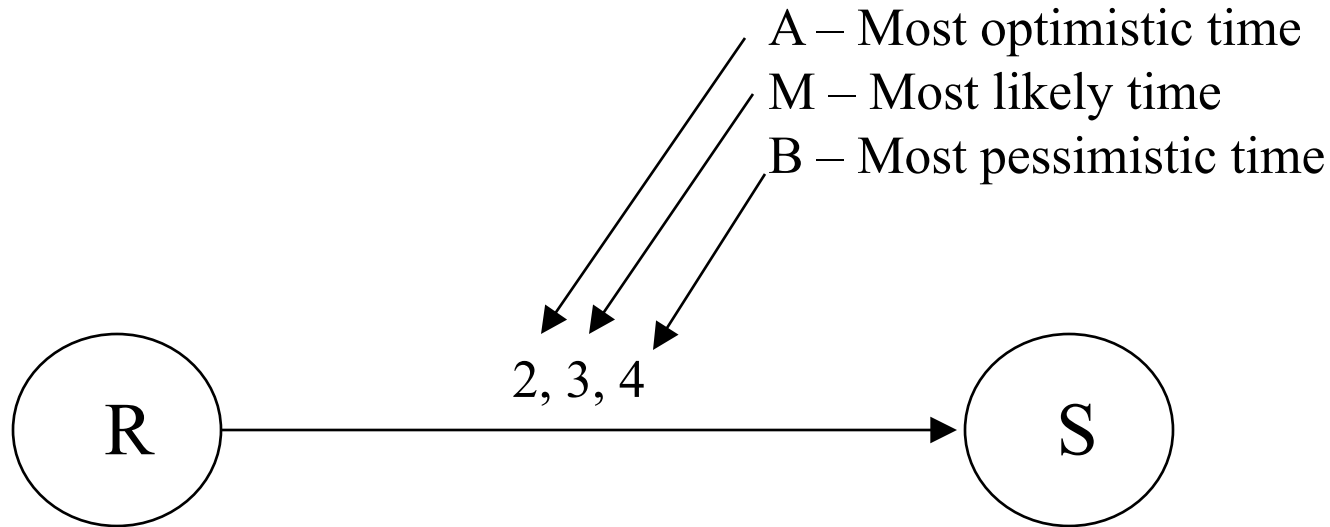
- or -

$$\text{Task slack time} = \text{LF} - \text{EF}$$

Task B has 2 time units of **slack time**

# PERT

---

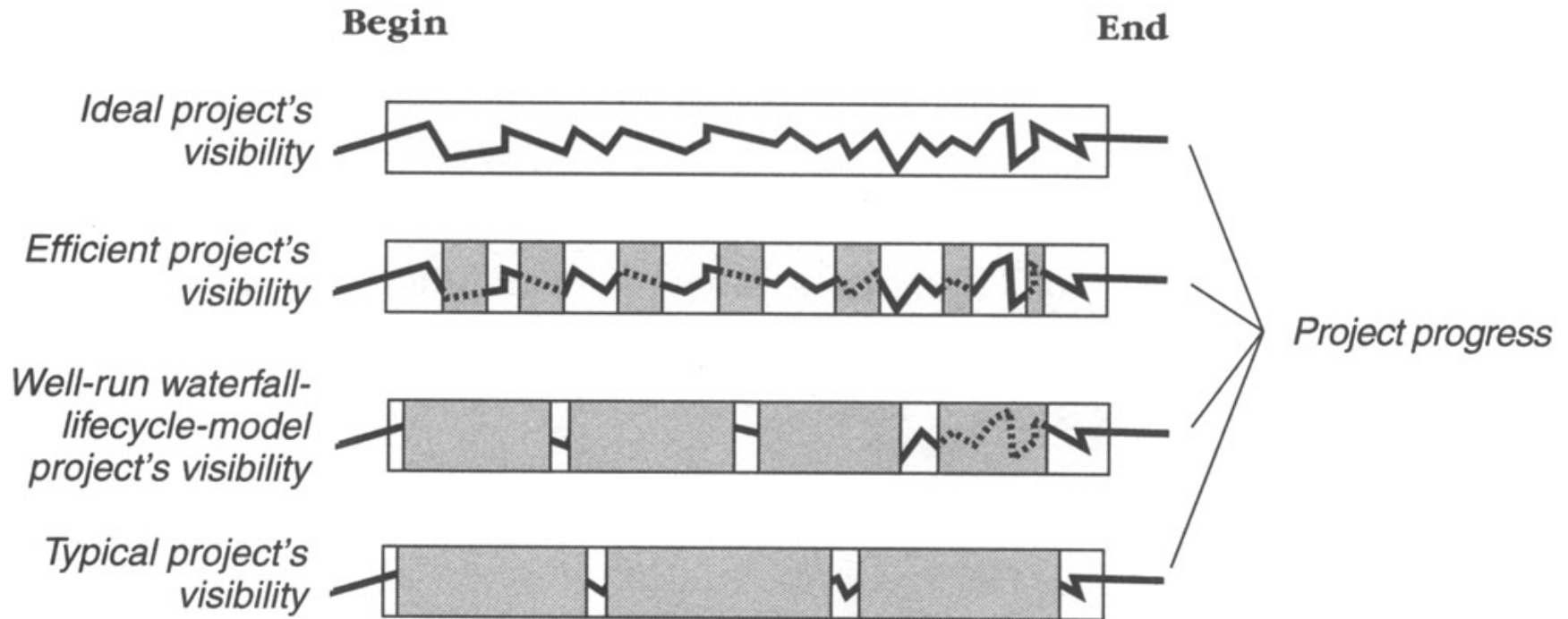


$$\text{Expected Time} = (a + 4m + b)/6$$

$$\text{Expected Time} = 3$$



# Tracking Visibility



# Percent Complete?

---

- 1) Conceptual Design – 200/200
- 2) Program Specification – 300/300
- 3) Coding – 150/600
- 4) Documentation – 10/100
- 5) User Manual Production – 0/400
- 6) Testing – 0/500 hours

$$660 / 2100 * 100 = 31.4\% \text{ complete}$$

