Estimation, Scheduling & Tracking

Week 4 – Spring 2026

Project Management

- Two over-arching inter-dependent aspects of software projects
  - Process
  - Project Management

  Main responsibilities of a project manager are
  - Project planning
  - Project monitoring and control

  (Major activities: Software Estimation, Scheduling and Tracking)

Estimate how long will it take for you to get to hostel from class today !

Estimations

- On what basis did you estimate?
  - Experience right? (History matters…)
  - Likely as an "average" probability'

Remember, an "exact estimate" is an oxymoron

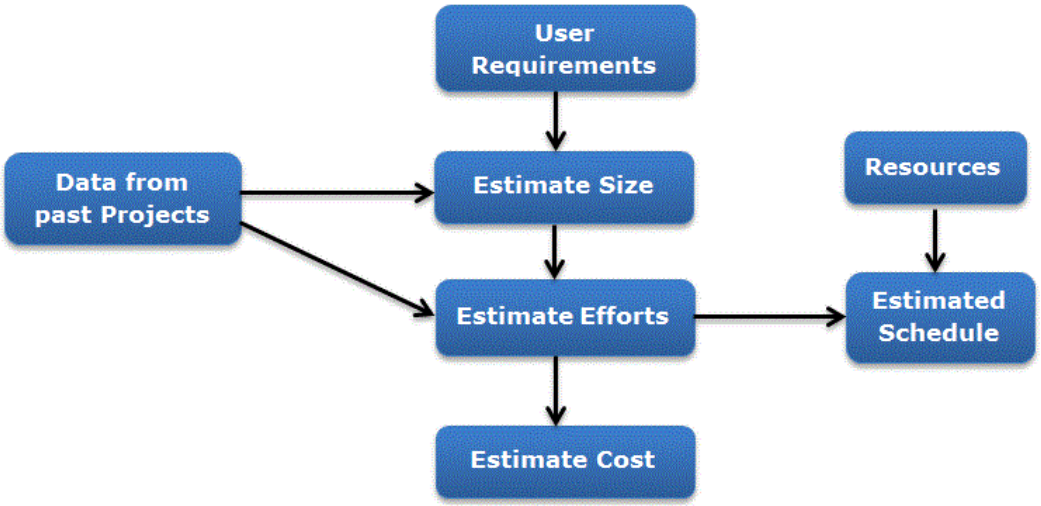
Can you estimate these?

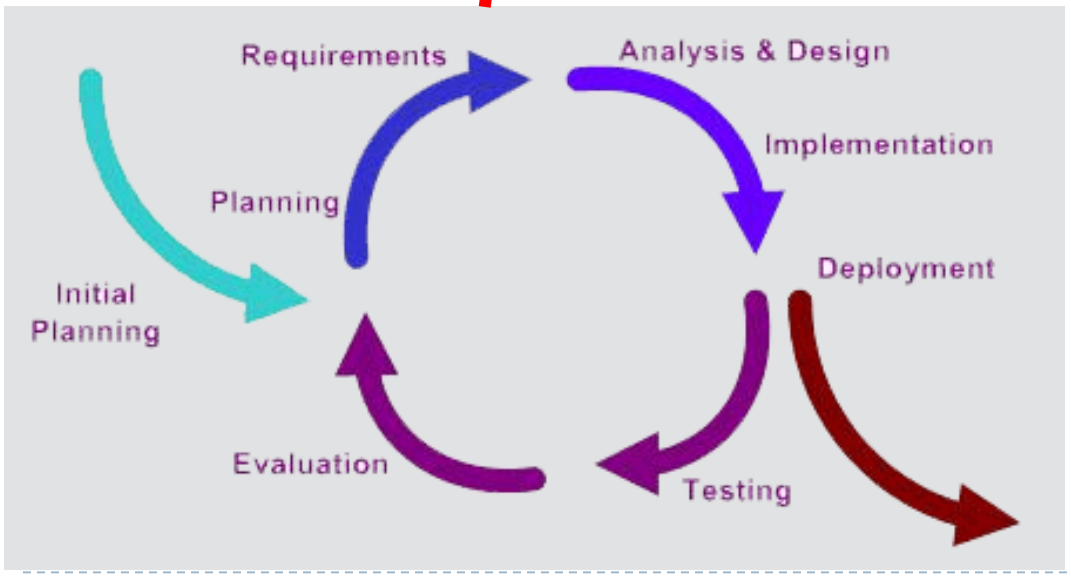
1. Surface temperature of the sun (in degrees C)
2. Latitude of Hyderabad (in degrees)
3. Surface area of Asia (in km^2)
4. Birth date of Alexander The Great (year)
5. Global revenue of "Titanic" (in $)
6. Length of the Pacific coastline (Ca, Or, Wa) (in km)
7. Weight of the largest whale (in tonnes)

1

Always give a range
Never give them a number

Approach


User Requirements
Data from past Projects
Estimate Size
Resources
Estimate Efforts
Estimated Schedule
Estimate Cost


Requirements
Analysis & Design
Implementation
Planning
Deployment
Initial Planning
Evaluation
Testing

#2

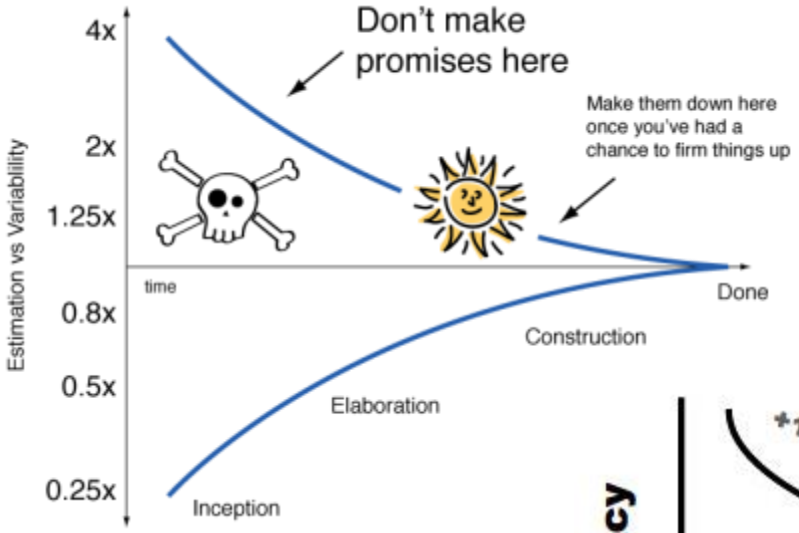
Always ask what the estimate will be used for

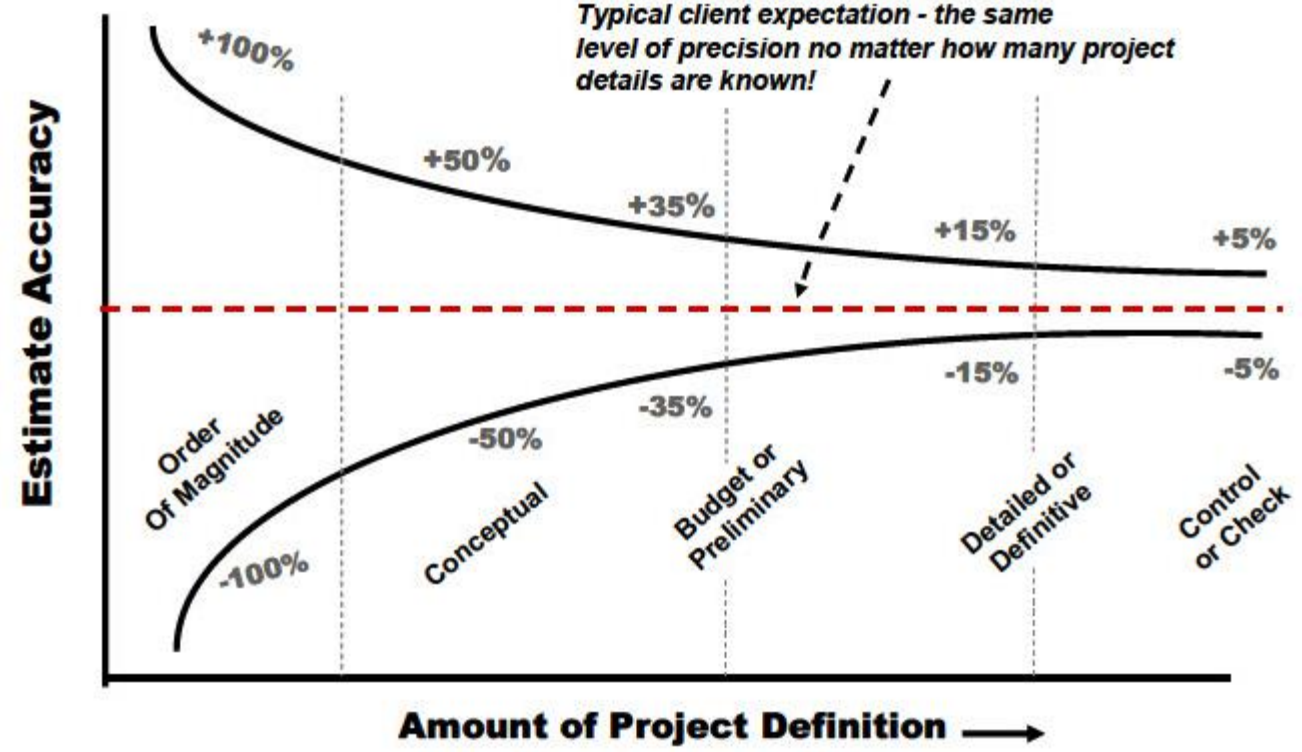
Requirements is key

#3

Estimation != Commitment

Iteratively increasing clarity



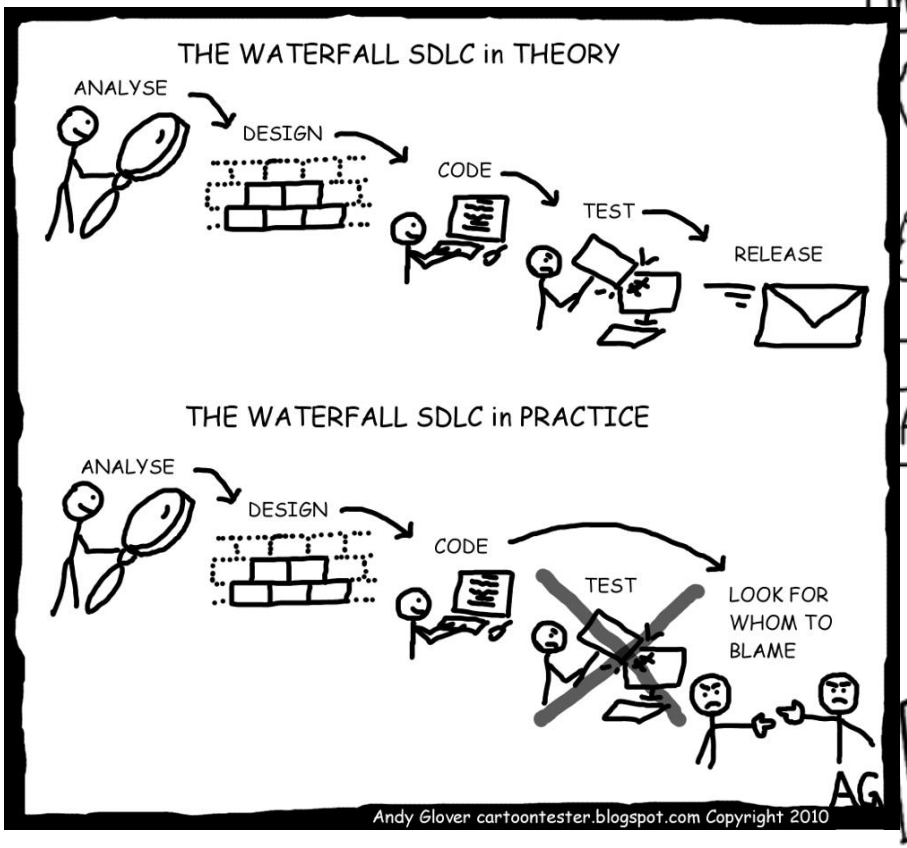


#4

First try to measure, count and compute

Estimate only when necessary

Reality


THE WATERFALL SDLC in THEORY
ANALYSE
DESIGN
CODE
TEST
RELEASE
THE WATERFALL SDLC in PRACTICE
ANALYSE
DESIGN
CODE
TEST
LOOK FOR WHOM TO BLAME
AG
Andy Glover cartoontester.blogspot.com Copyright 2010


Estimation
Inexperienced Developer
that'll take 2 days, tops.
Experienced Developer
that'll take at least 2 weeks.
Inexperienced PM
2 days? Sounds ok.
but I only scheduled a day and a half.
Experienced PM
2 weeks? Sounds ok.
It's going to be at least three.
AND THEN SALES COMES ALONG:
GOOD NEWS! I GOT THE CONTRACT SIGNED! 1 DAY!
Shit.

#5

Aggregate independent estimates

"Wisdom of the Crowds"

Team effort


Project Approval Board
Don't worry! My team is hard at work coming up with an accurate cost estimation for the project.
I say we each get 3 tries and then we average the results.
J. KING
MODERN ANALYST

The law of large numbers
(or: statistics is on our side, for once)

- If we estimate with an error of x%
- The estimate of each scope item will have an error of x%
- But...
- Some items will be over-estimated, others under-estimated (maybe....)

=> The error on the total estimate is likely < x%

Estimation Methodologies

- Top-down
- Bottom-up
- Analogy
- Expert Judgment
- Priced to Win (request for quote – RFQ)
- Parametric or Algorithmic Method
  - Using formulas and equations

Function Point Analysis (FPA) – Why and How

- Software size measured by number & complexity of functions it performs
- More methodical than LOC counts
- House analogy
  - House's Square Feet ~= Software LOC
  - # Bedrooms & Baths ~= Function points
  - Former is size only, latter is size & function
- Five basic steps

Function Point Analysis - Process

1. Count # of business functions per category
   - Categories: **outputs, inputs, inquiries, files or data structures, and interfaces**

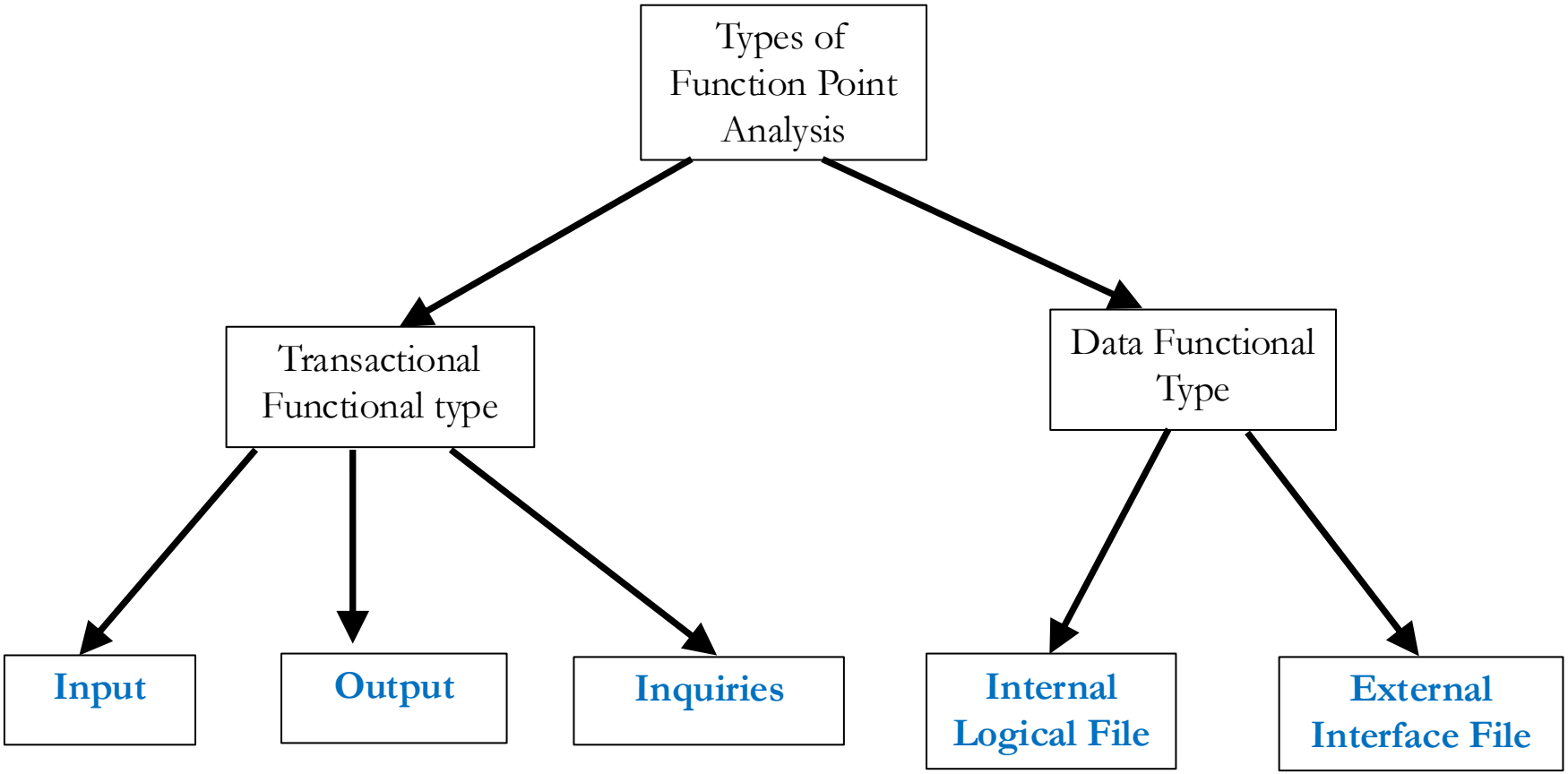
2. Establish Complexity Factor for each and apply
   - Low, Medium, High
   - Set a weighting multiplier for each (0 →15)
   - This results in the "unadjusted function-point total"

3. Compute an "influence multiplier"/ value adjustment factor (VAF) and apply
   - It ranges from 0.65 to 1.35; based on 14 factors

4. Results in "function point total"
   - This can be used in comparative estimates

5. Estimating effort and time
   - Calculate based on per function point effort

Function Point Analysis (FPA) - Categories



External Inputs (EI)

User/process inputs that create or update data

- Examples:
  - Data-entry screens/forms (e.g., "Create Customer", "New Order", "Submit Expense Report")
  - Uploads that create/modify records (CSV import of products)
  - API endpoints that add/update resources (POST /orders, PUT /user/profile)
  - Transactional commands (e.g., "Check in Book", "Return Item")

- Counting hints:
  - Count each elementary process that crosses the boundary and results in data creation/update.
  - Record Data Element Types (DET) = distinct user-recognizable fields on the input (e.g., OrderID, Quantity).
  - Record File Types Referenced (FTR) = number of files referenced/updated by the input.
  - Classify complexity (Low/Avg/High) using DET/FTR thresholds.

External Outputs (EO)

Outputs sent outside the application that contain calculations

- Typical examples:
  - Reports (monthly sales report, invoice generation)
  - Notifications containing computed values (e.g., overdue fines calculated)
  - API responses that include aggregated or derived values (e.g., account balance with interest calculation)

- Counting hints:
  - Count each distinct output that includes derived data or non-trivial processing.
  - DETs = data elements shown in the output.
  - FTRs = ILFs/EIFs referenced to produce the output.
  - EO usually has higher weight than EI for equivalent DET/FTR counts.

External Inquiries (EO)

Online queries that retrieve data with little or no processing

- Typical examples:
  - Search screens (search customer by ID, search book by title)
  - Single-record lookups (view customer profile)
  - Simple list displays without calculated fields (list of available copies)
  - Lightweight API GET that returns stored data without aggregation

- Counting hints:
  - Count each distinct inquiry that is a read-only retrieval with no significant derived values.
  - DETs = fields returned or input criteria for the query.
  - FTRs = ILFs/EIFs referenced by the query.
  - EQ weights are typically lower than EO since there is minimal processing.

Internal logical Files (ILF)

Application-maintained logical data groups

Typical examples:

Customer database table(s) maintained by the application
Orders, Products, Inventory, User Accounts within the system boundary
Configurations stored and updated by the system

Counting hints:

Count each ILF (logical file/table/group) maintained by the application.
DETs = distinct data elements (fields) stored in the ILF.
RETs = record element types (sub-groups/record types within the file, e.g., header/detail).
Classify complexity (Low/Avg/High) using RET & DET thresholds.

External Interface Files (EIF)

Read-only logical data used by the application but maintained externally

- Typical examples:
  - Read-only product catalog supplied by an external system
  - Shared master data stored in another application (e.g., central user identity store)
  - External configuration files, reference tables accessed but not modified (e.g., currency rates service cache)

- Counting hints:
  - Count each distinct external file or data group the application references.
  - Use DETs and RETs similarly to ILFs to determine complexity, but EIFs are weighted lower since they are not maintained by the system.
  - Only include if the application reads the external data; if it both reads and writes, consider it an ILF (if maintained within the application's boundary).

Example - Online Bookstore Application

- **User Management**: Registration, Login, Profile Update, etc.
- **Product Management**: Adding new books, Editing book details, etc.
- **Order Management**: Placing orders, Viewing order history, etc.
- **Search Functionality**: Searching for books, etc.
- **Reviews and Ratings**: Adding and viewing reviews, etc.

Step 1: Identify and Classify Functions

- Inputs - e.g., User Registration, Adding new books
- Outputs - e.g., Displaying order confirmation
- Inquiries - e.g., Searching for books
- Logical Files - e.g., User account details, Book details
- Interface Files- e.g., Interfaces with a payment gateway

Step 2: Assign Complexity Weights

For example (for inputs):

- Low complexity: 3 FP
- Average complexity: 4 FP
- High complexity: 6 FP

(for outputs):

- Low complexity: 4 FP
- Average complexity: 5 FP
- High complexity: 7 FP

Function point multipliers

	Function Points		
Program Characteristic	Low Complexity	Medium Complexity	High Complexity
Number of Inputs	x 3	x 4	x 6
Number of Outputs	x 4	x 5	x 7
Inquiries	x 3	x 4	x 6
Logical internal files	x 7	x 10	x 15
External interface files	x 5	x 7	x 10

From Rapid Development by Steve McConnell

Counting the Number of Function Points

Step 3: Calculate Unadjusted Function Points (UFP)
Step 4: Calculate Adjusted Function Points (AFP)

Influence multiplier/value adjustment factor (VAF) is based on 14 general system characteristics like data communications, distributed functions, performance requirements, etc.

Each characteristic is rated on a scale from 0 (no influence) to 5 (strong influence), and the total is summed up

	Function Points		
Program Characteristic	Low Complexity	Medium Complexity	High Complexity
Number of Inputs	5 x 3 = 15	2 x 4 = 8	3 x 6 =18
Number of Outputs	6 x 4 = 24	6 x 5 = 30	0 x 7 = 0
Inquiries	0 x 3 = 0	2 x 4 = 8	4 x 6 = 24
Logical internal files	5 x 7 = 35	2 x 10 =20	3 x 15 = 45
External interface files	8 x 5 = 40	0 x 7 = 0	2 x 10 = 20
Unadjusted function-point total		287	
Influence multiplier/value adjusted factor		1.20	
Adjusted function-point total		344	

From Rapid Development by Steve McConnell

Wideband Delphi

- Group consensus approach
- Present experts with a problem and response form
- Conduct group discussion, collect anonymous opinions, then feedback
- Conduct another discussion & iterate until consensus
- Advantages
  - Easy, inexpensive, utilizes expertise of several people
  - Does not require historical data
- Disadvantages
  - Difficult to repeat
  - May fail to reach consensus, reach wrong one, or all may have same bias

Scheduling & Tracking

Partitioning Your Project

- 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?

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

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.

Principles:

- Deliverable-oriented (focus on outputs, not activities)
- Hierarchical: levels from project → major deliverables → sub-deliverables → work packages → tasks
- Mutually exclusive elements (avoid overlap)
- 8–80 rule: work packages typically between 8 hours and 80 hours
- Include non-development work (requirements, QA, deployment, training, project management)

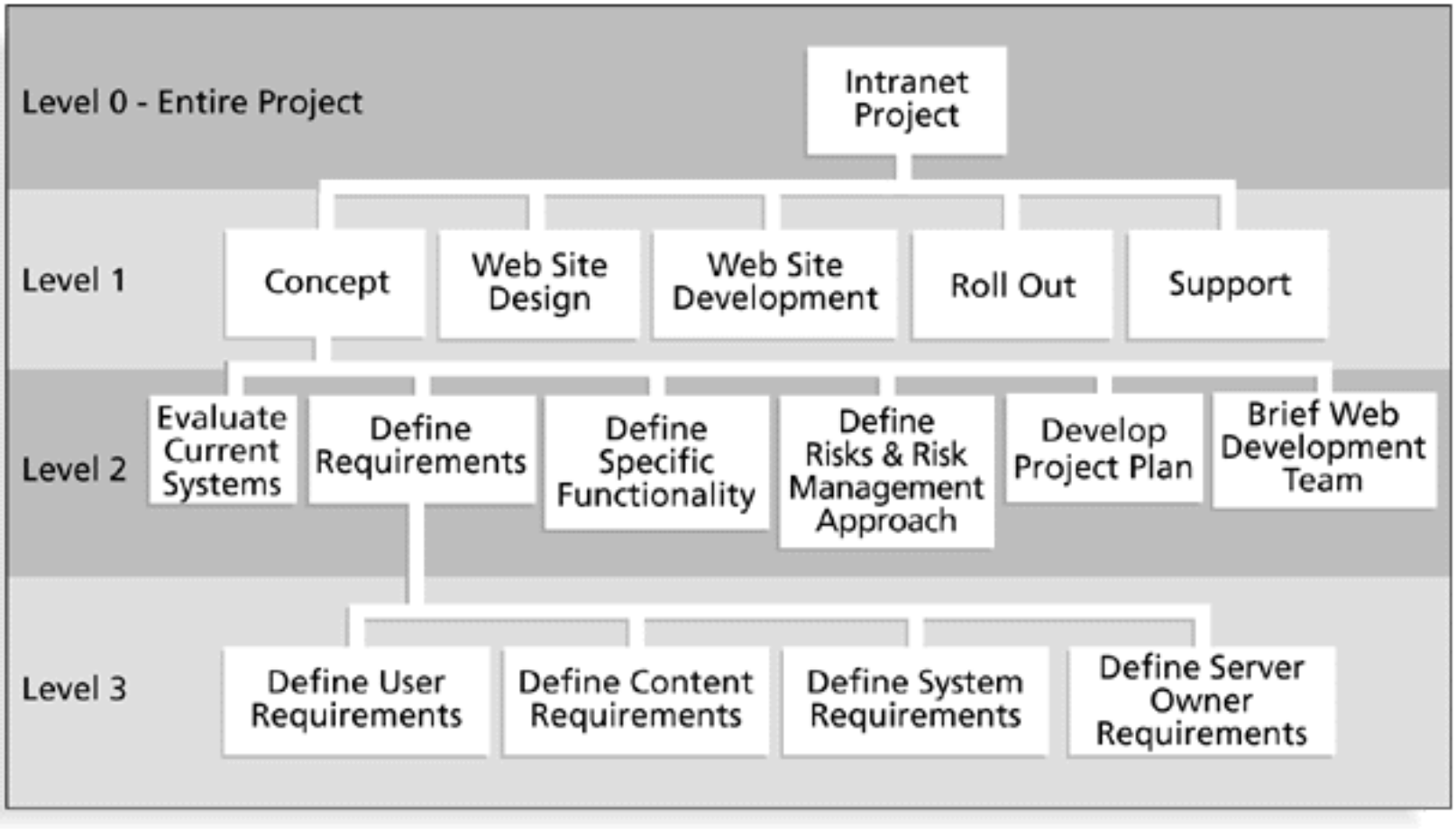
WBS Outline Example

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

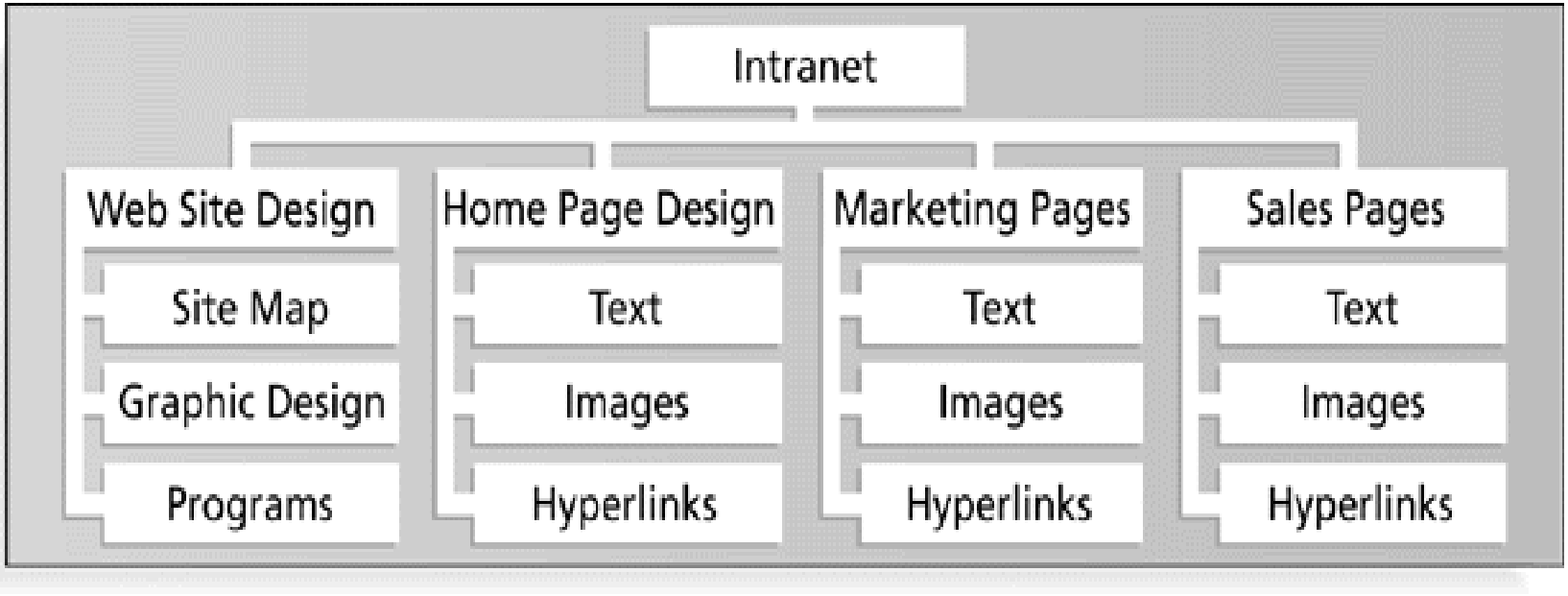
WBS Types

- 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

Process WBS


Level 0 - Entire Project
Intranet Project
Level 1
Concept
Web Site Design
Web Site Development
Roll Out
Support
Level 2
Evaluate Current Systems
Define Requirements
Define Specific Functionality
Define Risks & Risk Management Approach
Develop Project Plan
Brief Web Development Team
Level 3
Define User Requirements
Define Content Requirements
Define System Requirements
Define Server Owner Requirements

Product WBS


Intranet
Web Site Design
Site Map
Graphic Design
Programs
Home Page Design
Text
Images
Hyperlinks
Marketing Pages
Text
Images
Hyperlinks
Sales Pages
Text
Images
Hyperlinks

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
  - 1st pass: go 1-3 levels deep
  - Gather more requirements or data
  - Add more detail later
- Post-its on a wall

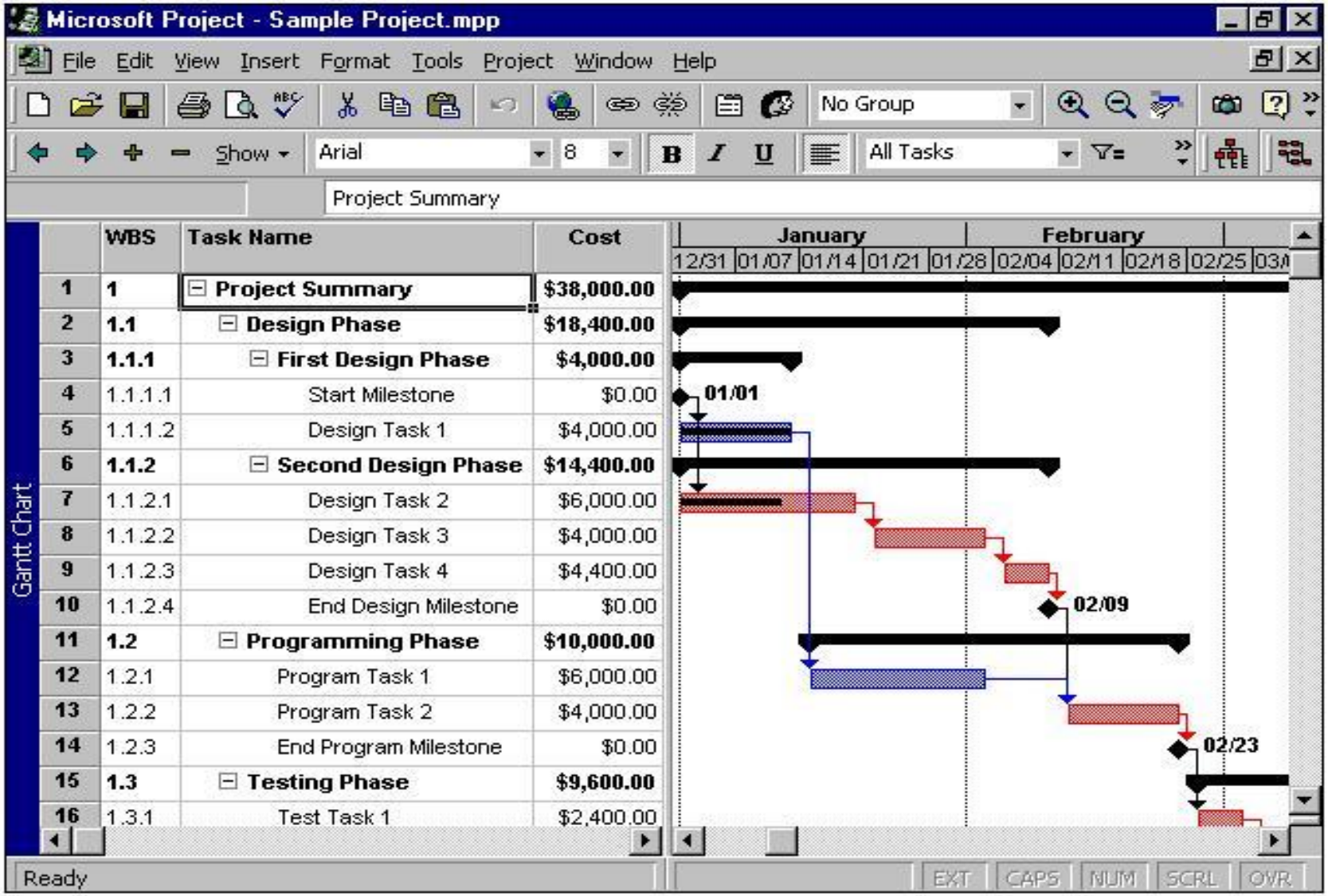
Thank you

Scheduling Activities

- 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 cost
- Most commonly used for exhibiting project 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


Microsoft Project - Sample Project.mpp
File Edit View Insert Format Tools Project Window Help
No Group
Show
Arial
8
All Tasks
Project Summary
WBS
Task Name
Cost
1 1 Project Summary $38,000.00
2 1.1 Design Phase $18,400.00
3 1.1.1 First Design Phase $4,000.00
4 1.1.1.1 Start Milestone $0.00
5 1.1.1.2 Design Task 1 $4,000.00
6 1.1.2 Second Design Phase $14,400.00
7 1.1.2.1 Design Task 2 $6,000.00
8 1.1.2.3 Design Task 3 $4,000.00
9 1.1.2.3 Design Task 4 $4,400.00
10 1.1.2.4 End Design Milestone $0.00
11 1.2 Programming Phase $10,000.00
12 1.2.1 Program Task 1 $6,000.00
13 1.2.2 Program Task 2 $4,000.00
14 1.2.3 End Program Milestone $0.00
15 1.3 Testing Phase $9,600.00
16 1.3.1 Test Task 1 $2,400.00
January
February
12/31 01/07 01/14 01/21 01/28 02/04 02/11 02/18 02/25
01/01
02/09
02/23
Gantt Chart
Ready
EXT CAPS NUM SCRL OVR

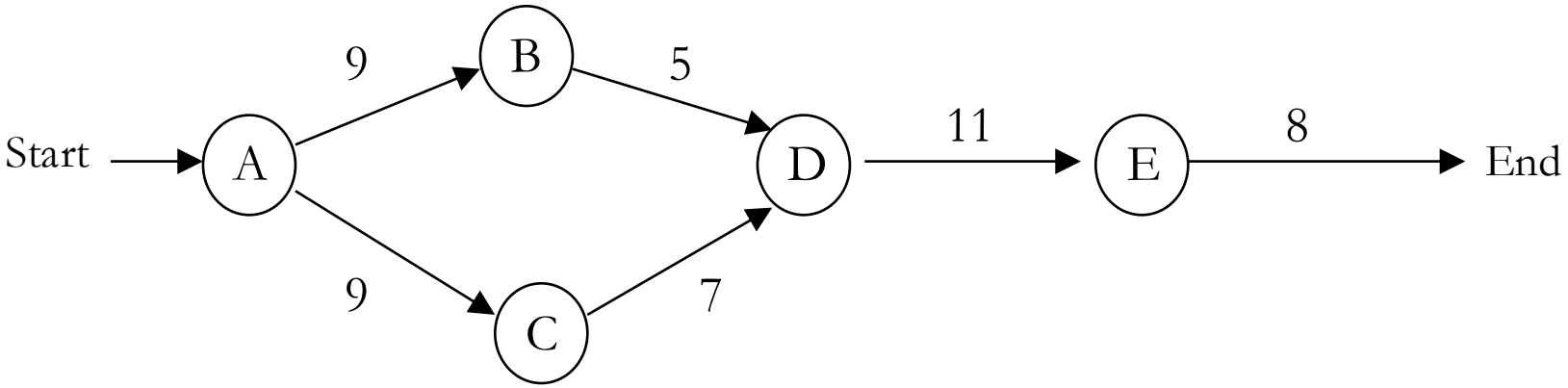
Gantt Chart

- 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



Critical Path Method (CPM)

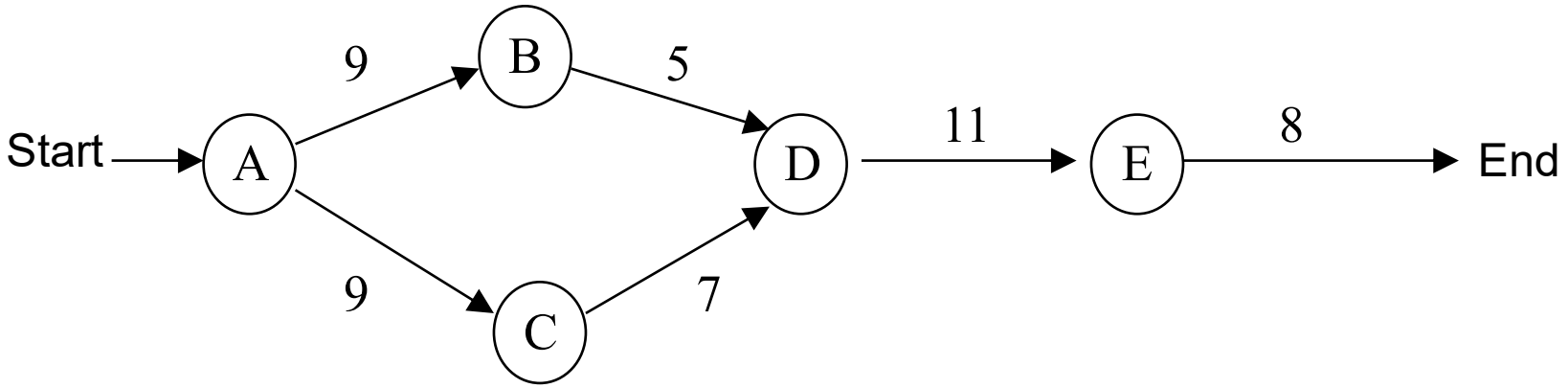
CPM tries to answer the following questions:

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

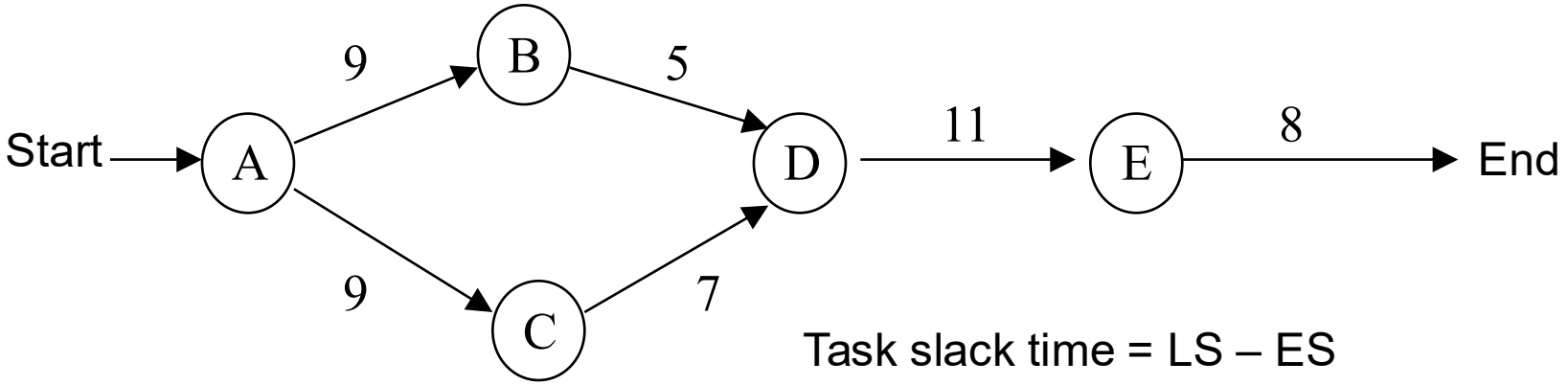


Critical Path = A – C – D – E (35 time units)

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.

Task slack time = LS – ES

- or -

Task slack time = LF - EF

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

Integrating Estimation with Scheduling and Tracking

- Example:
  - FPA yields 143 FP → 2860 hours at 20 hours per FP.
  - With a team of 5 devs (150 hours/week), timeline: 2860/150 ≈ 19 weeks.
  - Add buffers (~20%) for uncertainties → ~23 weeks total.

- Monitoring progress:
  - Use metrics like Earned Value Management (EVM):
    - Planned value (PV), earned value (EV), actual cost (AC).
    - Example: After 4 weeks, PV = $40K, EV = $35K, AC = $38K.
    - Calculate CPI and SPI to assess schedule and cost performance.

Summary

- Estimation is foundational for accurate scheduling.
- Scheduling tools (Gantt, PERT, CPM) make plans visible and manageable.
- Use tracking metrics (EV, CPI, SPI) to control projects effectively.