# Project Management

Week 4- Project Planning and Network Diagrams

# Last Week

- Project Appraisal Techniques
- Scoping Tools-Breakdown Structures
- Estimation

# This Week

- What is Planning?
- Purpose of Planning
- Planning Process
- Estimating Task Duration
- Task Dependency
- Network Analysis/Critical Path Analysis

# ‘The Plan is nothing, Planning is everything’

Lord Nelson and General Eisenhower

# Planning

- It is about **sequencing** project activities and milestones into a sensible and logical order
- It involves linking activities to form a project network to show how different activities are related to each other
- One requirement is to establish the **timing** and resources

# Purpose of Planning

- Its is used to find the expected project duration and milestone dates
- It is used to make sure that the project is completed by a target date
- To get stakeholder commitment

# The Planning Process

Identify Activities

Determine Sequence of activities

Prepare Estimates of Time and Resources

Present Plan in a readily intelligible format

# Estimating Task Duration

- Use Historical Data- past experience is a good predictor of future events
- Time the activity- possible to do a trial run
- Use a probabilistic method-  such as expert judgment or simulation techniques or Weighted Average

# Weighted Average

- Most Optimisitc (a)- the minimum time
- Most Likely (m)-  the most likely duration assuming normal conditions
- Most Pessimistc (b)- the expected duration if major difficulties are encountered

Example

- Most Optimistic (a) = 24hrs
- Most Likely (m) = 48hrs
- Most Pessimistic (b) = 96hrs

- Weighted Average

$= (a + 4m + b)/6$

$= (24 + 4(48) + 96)/6$

$= 312/6$

$= 52$ hrs

Using a distribution called a beta distribution approximation

# Tasks Dependencies

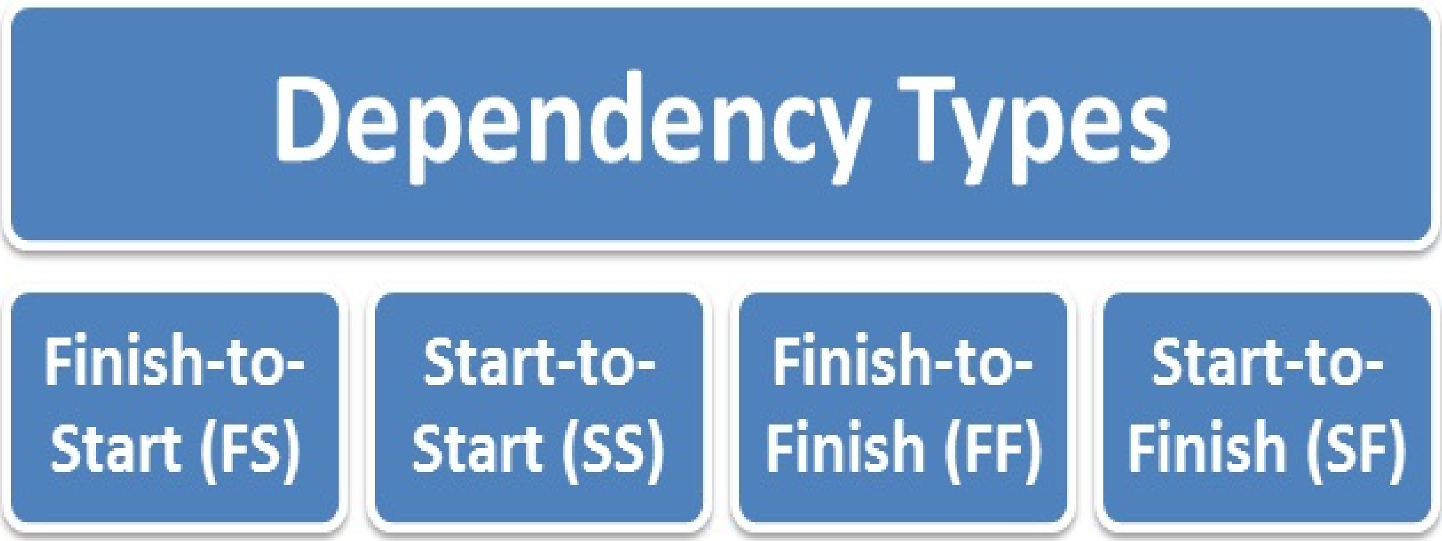
- A relationship between two linked tasks
- Linked by a dependency between finish and start dates
- Defining task dependencies correctly results in a schedule that's easy to maintain regardless of what happens or how quickly
- There are four types of dependency: Finish to Start, Start to Start, Finish to Finish and Start to Finish

# Task dependencies vs Date Constraints

- Applying dependencies to tasks is preferable to setting hard dates in a schedule.
- Fixed dates applied to tasks, called date constraints, quickly increase the effort to maintain the schedule.
- If you use date constraints, you often must manually recalculate dates when the schedule changes.
- However, date constraints don't always cause schedule problems.
- For example, when tasks occur on specific dates, such as training classes or conferences, date constraints keep those events tied to the correct dates on the calendar.

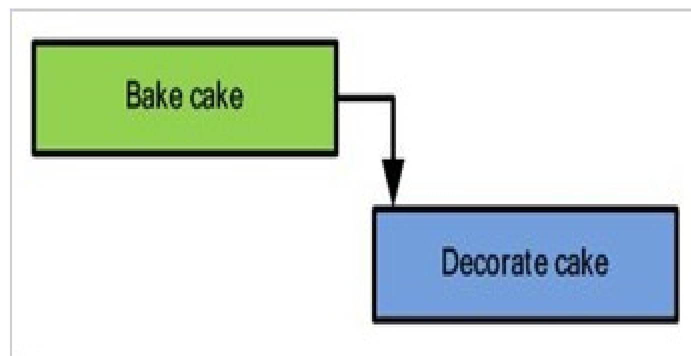
# Chronology vs Control

- Although tasks are called either successors or predecessors, a dependency is not really about chronology, but about control.
- A task dependency specifies which of two tasks controls the scheduling of the other.
- The independent task (the predecessor) determines the scheduling of the dependent task (the successor).
- A predecessor can occur before, at the same time as, or after its successor.


Dependency Types
Finish-to-Start (FS)
Start-to-Start (SS)
Finish-to-Finish (FF)
Start-to-Finish (SF)

# Finish-to-start (FS)

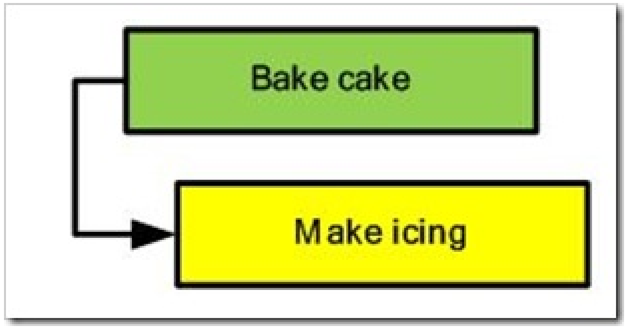
- This type of dependency is the most common, perhaps because control and chronology work in the same direction in this dependency. After the predecessor task finishes, the successor task starts.
- This means activity A must finish before activity B has permission to start




- For example:
  - After members of a construction crew set up the concrete forms for a foundation, they start to pour the concrete into the forms.

# Start-to-start (SS)

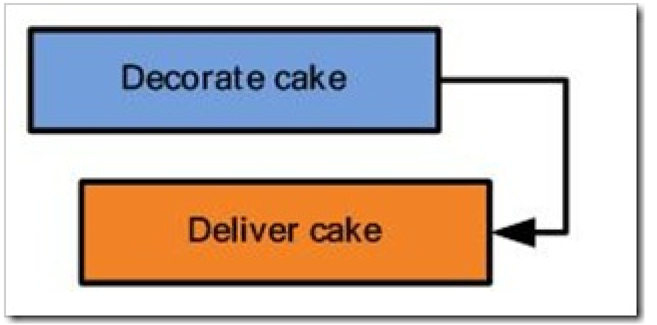
- This type of dependency indicates that the start of one task triggers the start of the second task. A lag time between tasks is often used with this type of dependency.
- This means that once activity A has started, activity B can also start.




- For example:
  - Members of a road crew start to place traffic cones to close a lane on the highway. Ten minutes (lag time), after they start the line-painting machine starts to paint lines.

# Finish-to-finish (FF)

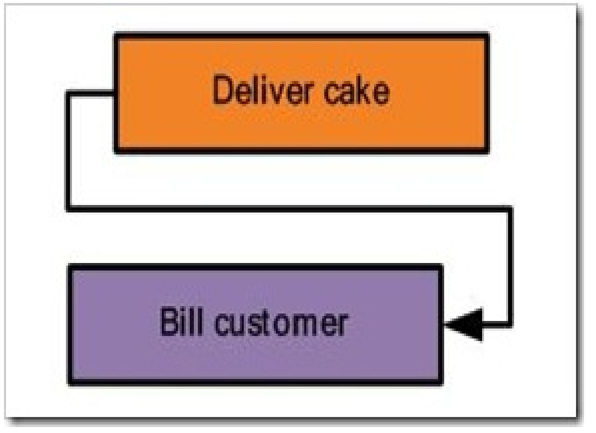
- This type of dependency includes one task that continues only as long as another task is in progress.
- This means activity A must finish before activity B can finish




- For example:
  - Traffic flaggers direct traffic until construction work is completed.
  - The concession stand at a sports arena stops serving refreshments when the game ends. In fact, it often stops serving alcoholic beverages two hours before the game ends (but not in Trinidad)

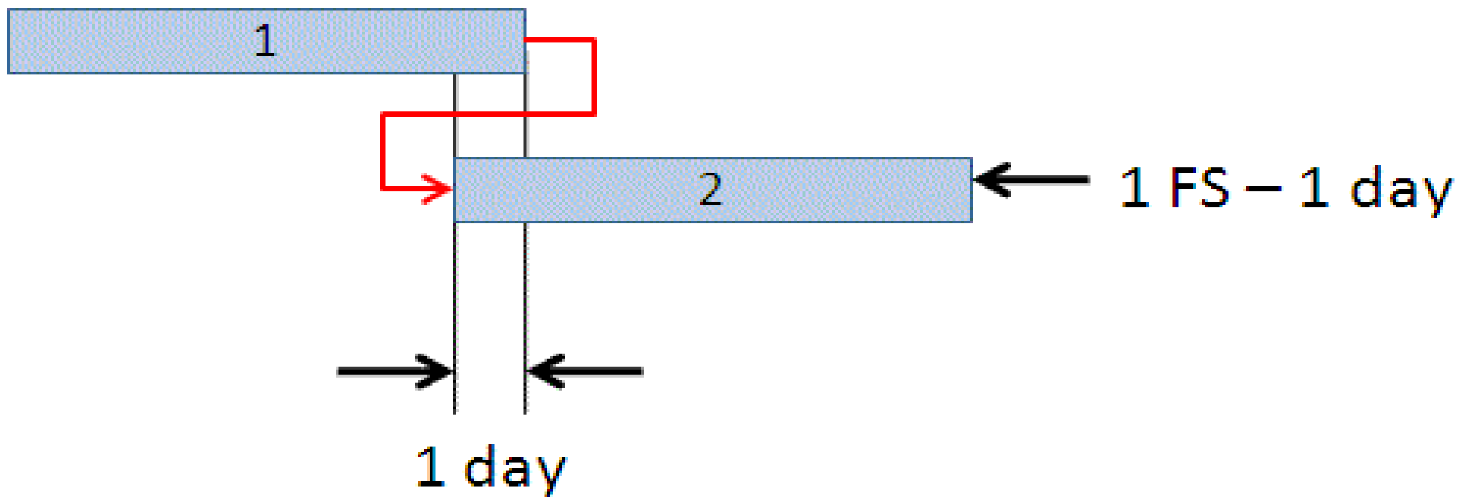
# Start-to-finish (SF)

- This type of dependency turns most people's concept of predecessor and successor tasks upside down. The confusion occurs because in most start-to-finish cases, the predecessor occurs after the successor. And the start of the predecessor controls the finish of the successor.

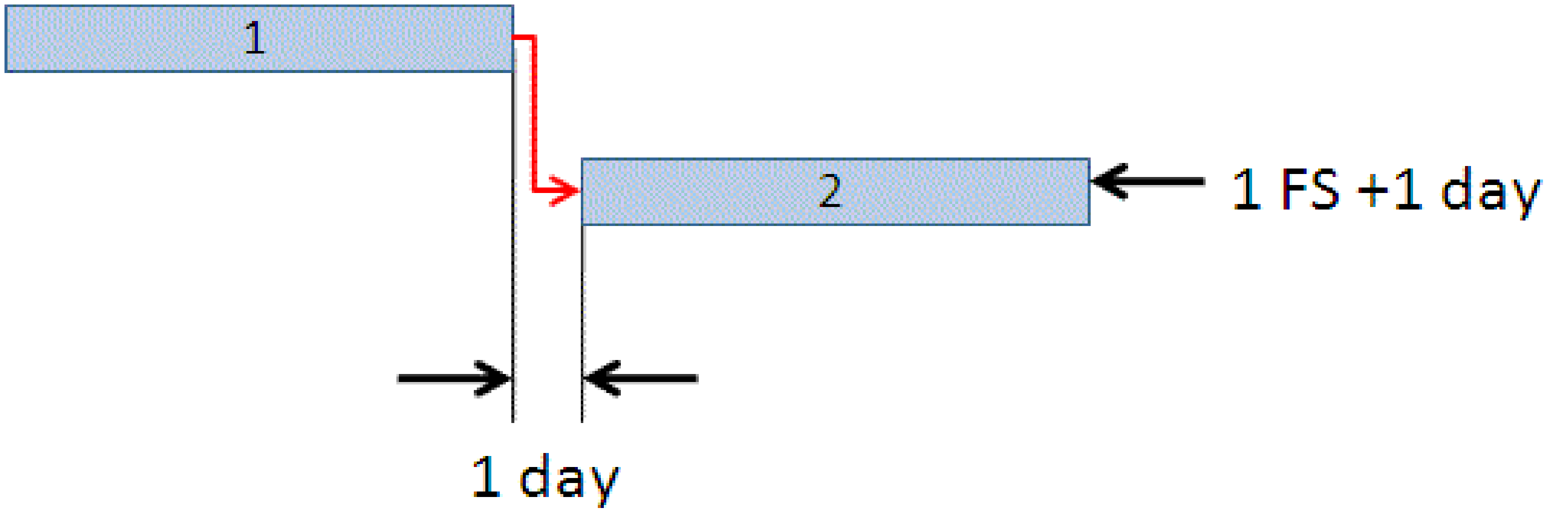



- For example:
  - Conference registration must start whether the registration booth is ready or not. Therefore, the start of the task "Check in registered attendees" controls the end of the task "Set up registration booth."

- **Lead Time-** is overlap between tasks that have a dependency.
  - For example, if a task can start when its predecessor is half finished, you can specify a finish-to-start dependency with a lead time of 50 percent for the successor task. You enter lead time as a **negative value** in MS Project.



- **Lag Time-** is a delay between tasks that have a dependency.
  - For example, if you need a two-day delay between the finish of one task and the start of another, you can establish a finish-to-start dependency and specify two days of lag time. You enter lag time as a **positive val**ue in MS Project

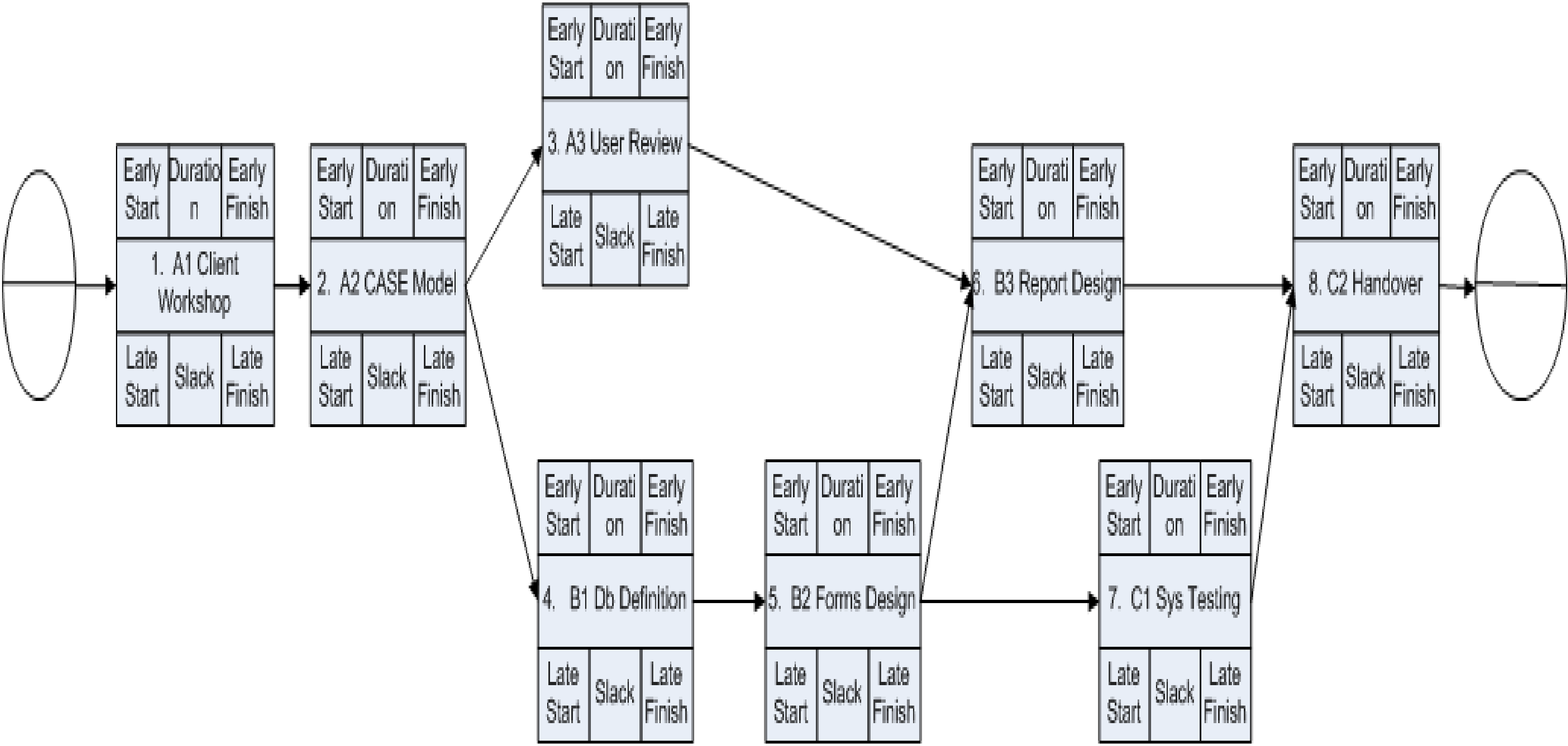


# Example using A-O-N

| ID | TASK NAME | DURATION | PREDECESSORS |
|---|---|---|---|
| 1 | A1 CLIENT WORKSHOP | 10 | - |
| 2 | A2 CASE MODEL | 15 | 1 |
| 3 | A3 USER REVIEW | 10 | 2 |
| 4 | B1 Db DEFINITION | 10 | 2 |
| 5 | B2 FORMS DESIGN | 20 | 4 |
| 6 | B3 REPORT DESIGN | 5 | 3,5 |
| 7 | C1 SYSTEM TESTING | 20 | 5 |
| 8 | C2 HANDOVER TO CLIENT | 5 | 6,7 |

| EST | DUR | EFT |
|---|---|---|
| TASK NAME | | |
| LST | FLOAT | LFT |

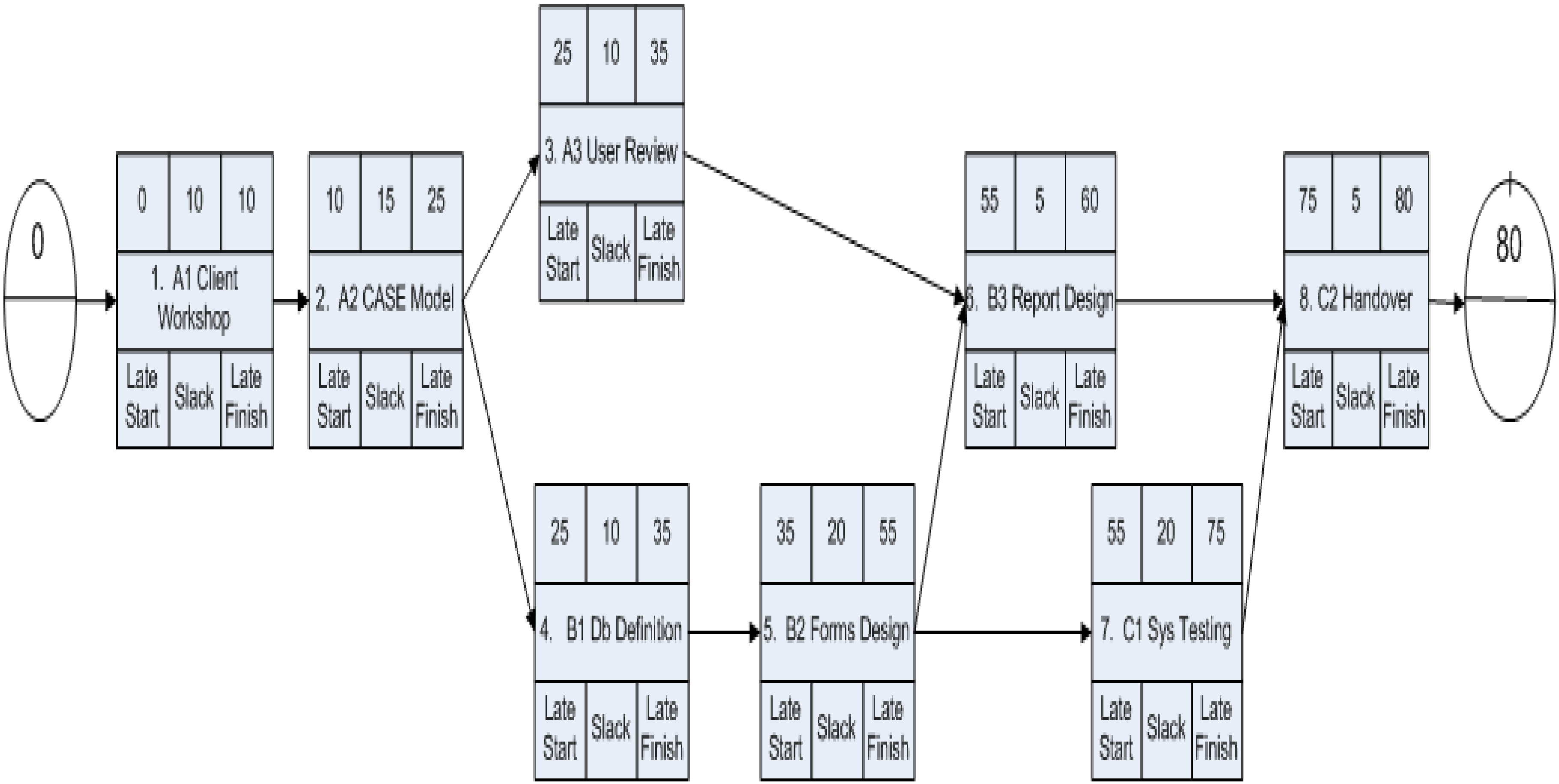
# Network Diagram



# **Forward Pass** to calculate EST and EFT

- The EST for the first activity is zero
- EFT for an activity is always found by adding its duration to its EST. i.e. **EFT = EST + Duration**
- The EST for all remaining activities is the same as the EFT of its immediate predecessor i.e. **EST = EFT of preceeding activity**
- In the case of convergence, **the EST is taken from the path having the highest EFT**
- The EFT of the last activity is the duration of the project

# Forward Pass



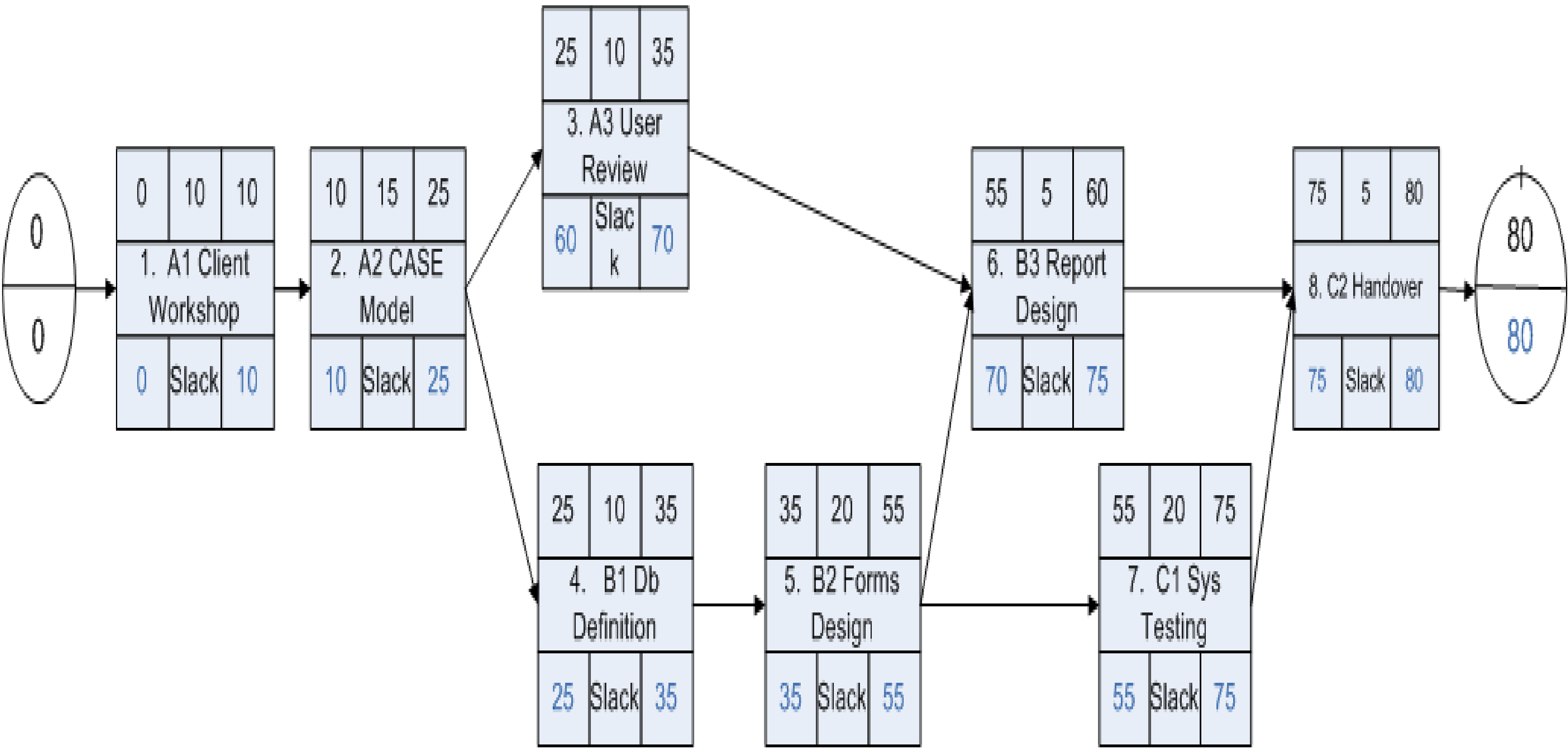
# **Reverse Pass** to calculate LST and LFT

- The LFT for the last activity is the same as its EFT
- The LST for an activity is always found by subtracting its duration from its LFT i.e.

  **LST = LFT – Duration**
- The LFT for all remaining activities is given by the LST of its immediate successor i.e.

  **LFT = LST of the successor activity** (moving from start to finish)
- In case of convergence, **the LFT is taken from the path having the lowest LST**

# Reverse Pass



# Float/Slack

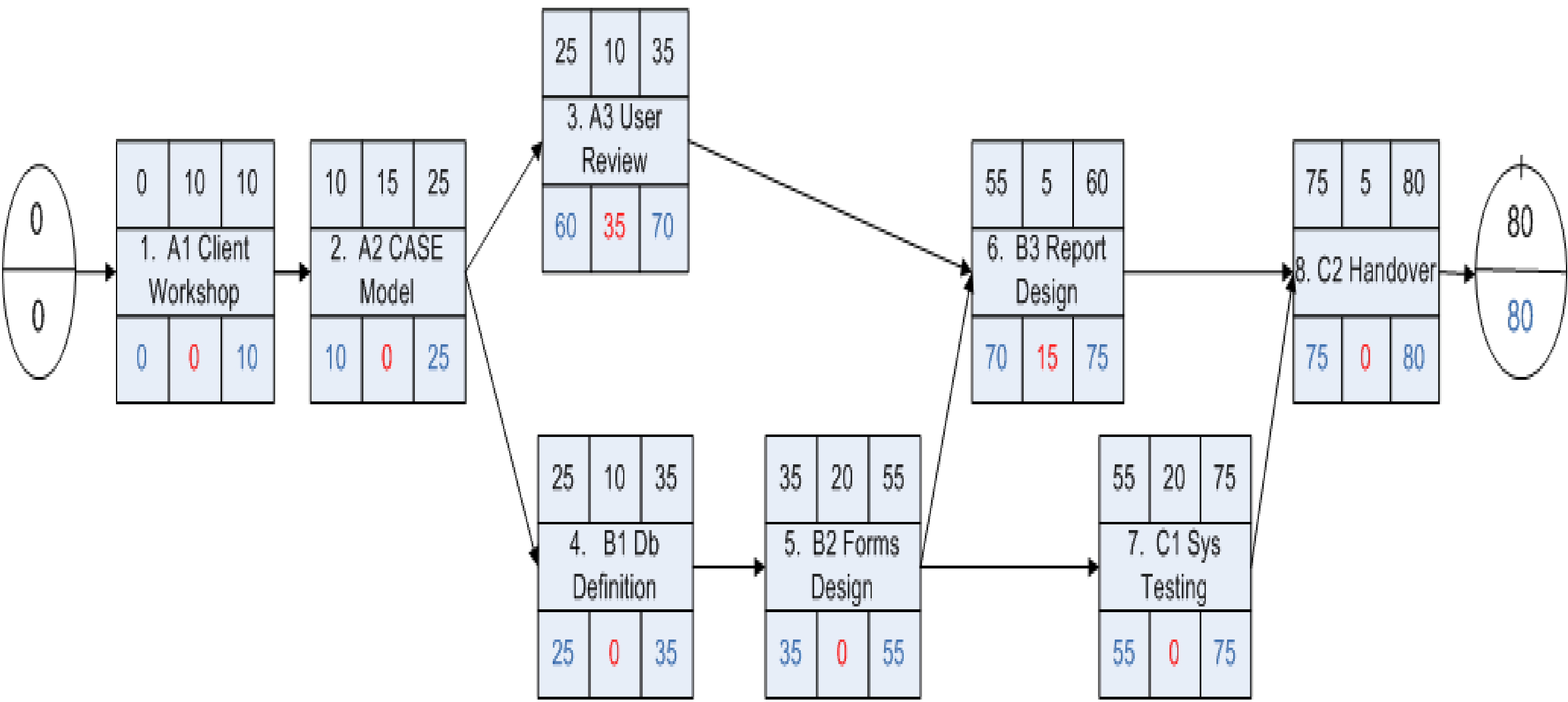
- A critical task is one where EFT= LFT.  Thinking this through it means that the activity MUST start on that date , the EST or the entire project duration is affected
- **Simply put an activity is Critical when the Float = 0**
- A non-critical task is where EFT < LFT.  This means that although the activity could start as early as the EST, providing it finishes by the LFT the project could still finish on schedule

- **Float/Slack** = LST – EST

  OR = LFT - EFT

- Critical activities must stay on schedule: knowing the float for non-critical activities will let the project manager know how long they can be delayed for before impacting the entire project.

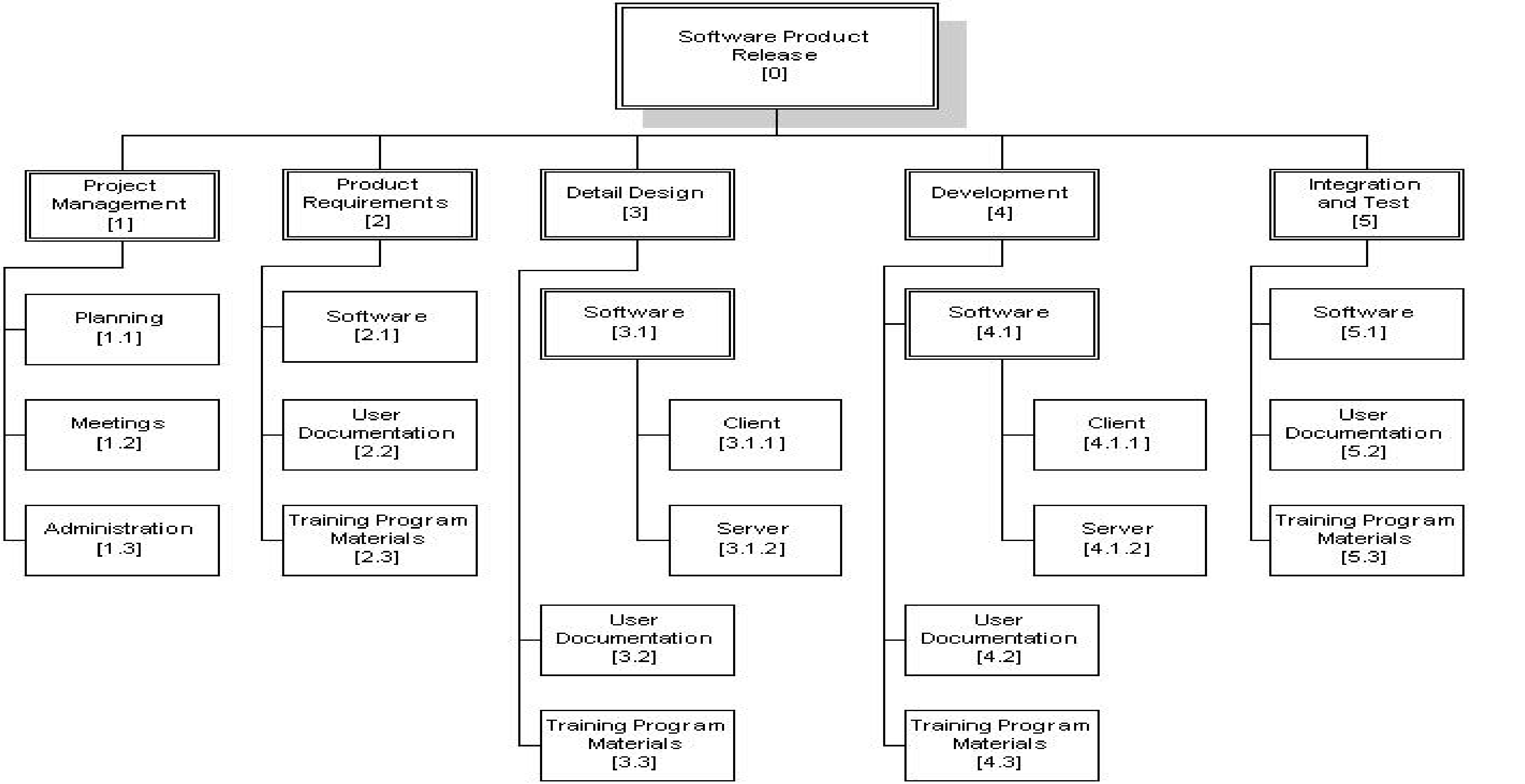
# Float/Slack



0
0
0
10
10
1. A1 Client Workshop
0
0
10
10
15
25
2. A2 CASE Model
10
0
25
25
10
35
3. A3 User Review
60
35
70
25
10
35
4. B1 Db Definition
25
0
35
35
20
55
5. B2 Forms Design
35
0
55
55
5
60
6. B3 Report Design
70
15
75
55
20
75
7. C1 Sys Testing
55
0
75
75
5
80
8. C2 Handover
75
0
80
80
80

- **Critical Path-** The critical path is the series of tasks (or even a single task) that dictates the calculated finish date of the project.
- That is, when the last task in the critical path is completed, the project is completed.
- It is identified by the series of task where the Float or Slack is ‘Zero’

# Example WBS



Software Product Release [0]
Project Management [1]
Planning [1.1]
Meetings [1.2]
Administration [1.3]
Product Requirements [2]
Software [2.1]
User Documentation [2.2]
Training Program Materials [2.3]
Detail Design [3]
Software [3.1]
Client [3.1.1]
Server [3.1.2]
User Documentation [3.2]
Training Program Materials [3.3]
Development [4]
Software [4.1]
Client [4.1.1]
Server [4.1.2]
User Documentation [4.2]
Training Program Materials [4.3]
Integration and Test [5]
Software [5.1]
User Documentation [5.2]
Training Program Materials [5.3]