

## TIME MANAGEMENT

## Time Management begins with the WBS

- We need a Work Breakdown Structure to get a list of all tasks and how long they will take.
- We can't manage anything without a WBS.

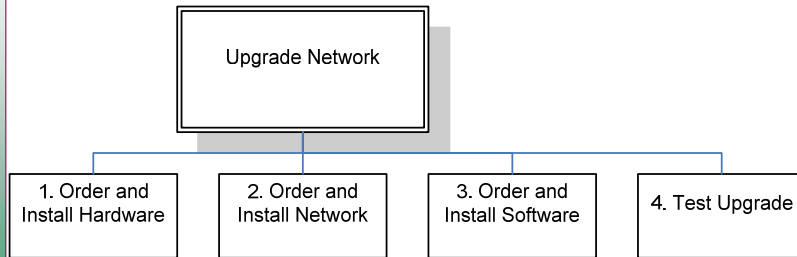
## WBS Recap:

- A WBS breaks the project down into smaller and smaller detail.
- The smallest level of detail are **work packages**.
- A Work Package is:
  - A manageable work effort
  - Assigned to a specific individual (rarely more than 1 person)
  - Will be completed in less than a week

## A WBS Example

- We are going to upgrade the network, workstations and software in an office.
- We start by listing the high level tasks:
  - Order and install workstations
  - Order parts for network and install
  - Order and install software
  - Test upgrades

- Let's show this as a diagram:
  - (The numbers represent the level of WBS)

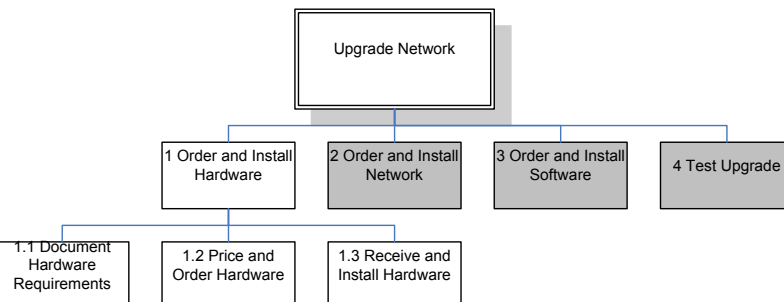


## WBS Example (2)

- Is this enough detail?
  - NO!**
    - What's involved in ordering and installing hardware? Who does it? How long does it take?
    - Similarly, what's involved in ordering and installing network, software? Who does it? How long does it take?
    - How do we test? Who does the testing?

## WBS Example (3)

- Let's just take one item from our Level 1 list: Order and Install Hardware
- It has at least 3 separate activities.

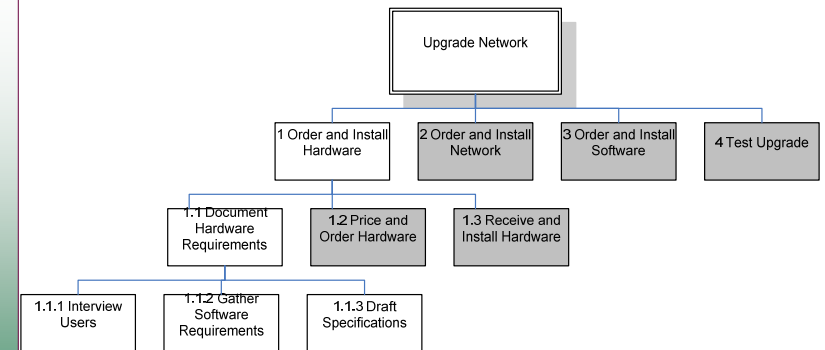


## WBS Example (4)

- Is this enough detail?
  - Will the same person do the work in each box?
  - Do we know how long each box will take to complete?
  - Will each box take less than a week to complete?
- If any of these answers is "No", then we need to further decompose the task.

## WBS Example (5)

- Maybe Document Requirements will take more planning and investigation:
  - Someone will interview users to understand their needs.
  - Someone will research hardware requirements to make sure the new machines are properly sized for the software.
  - Somebody has to put this together into a specification.



## WBS Example (6)

- We're at the point where we can say that a team member will take 4 hours to interview users
- Another team member will take 2 hours to go through every software package and add up what it needs for hardware.
- Somebody then takes 2 hours to put them into a specification document.

## WBS Example (7)

- We found the detail we need at WBS Level 3.
- Now we do this for the rest of the WBS (you can do that on your own).

## Milestones

- We use a special type of activity called a Milestone.
- **A Milestone** is an event with no duration. It has no duration because it takes no (or little) work.
- Milestones are used to signify important events in the life of a project.

- Sponsors are usually very interested in milestones.
- Milestone examples include:
  - Signing a contract or agreement
  - Groundbreaking for a new building
  - Completion of the implementation phase
  - Handover to the client

## The WBS Complete

- There are many ways to record the final WBS:
  - Microsoft Project
  - An Excel Spreadsheet
  - A Word Document
  - Network Diagrams
- Let's use a Spreadsheet Table:

WBS	Description	Effort	Task Assigned to:
1	<b>Order and Install Hardware</b>	--	
1.1	<b>Document Hardware Requirements</b>	--	
1.1.1	Interview Users	4h	Mariam
1.1.2	Gather Software Requirements	2h	Ahmed
1.1.3	Draft Specifications	2h	Mariam
1.2	<b>Price and Order Hardware</b>		
1.3	<b>Receive and Install Hardware</b>		
2	<b>Order and Install Network</b>	--	
...	...		...
3	<b>Order and Install Software</b>	--	
...	...		...
4	<b>Test Upgrade</b>	--	
...	...		...

## Dependencies

- Now that we have our WBS, we need to think about it a bit:
  - What tasks must be completed first?
  - What tasks depend on other tasks to be done?
  - Are there tasks that must be done on a certain date (and not before)?
  - Are there certain tasks that must be done by a certain date (and not later)?

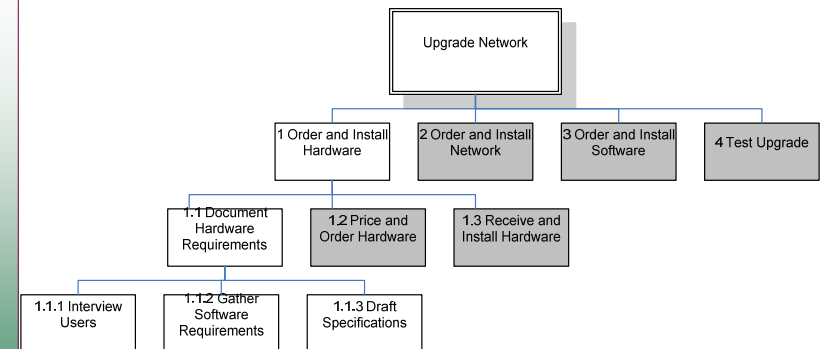
- A **Dependency** is a relationship where one task depends on another task/event before it can start or finish
- Dependencies come in 4 types:
  - Start-Finish
  - Start-Start
  - Finish-Start
  - Finish-Finish

## Dependencies Example

- **Start-Finish:** I can only start to install the furniture when you are finished telling me where to put it. (most common)
- **Finish-Finish:** I can only finish wiring the network when you finish installing the switches. (next common)
- **Start-Start:** I can only start to carry this heavy box when you start to carry it with me.
- **Start-Finish:** I can only finish my delivery when you start to unload the truck. (very uncommon)

## Dependency Examples

- Given our example:



- We can't Draft Specifications until we are finished Interviewing Users and Gathering Software Requirements.
- Therefore Draft Specifications has a Start-Finish dependency on these other two activities.

## Mandatory, Discretionary and External Dependencies

- **Mandatory Dependencies** are those that must be obeyed.
  - Eg. We can't put the roof on a building until we finish the walls.
- **Discretionary Dependencies** are "nice to have." We don't normally pay attention to these.
  - Eg. Let's wait until the computers arrive before we figure out the furniture for them.
- **External Dependencies** are enforced by external conditions.
  - Eg. Y2K was a big external dependency.

## Lag Times

- Some tasks take a certain time to complete once started. Nothing can be done until the lead or lag time is complete:
  - A baby has a 9 month lag time before you need to buy diapers.
  - Cement has a lag time of a day or two before you can walk on it.

## Lead Times

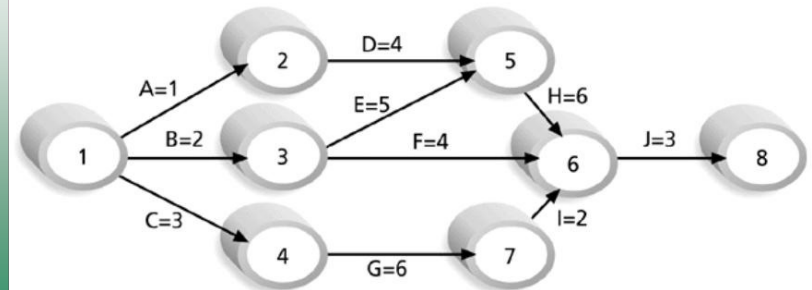
- Lead Times are a "hurry up". We can start on a task early but only if its preceding tasks are well underway.
  - Eg. It might take 4 weeks to plaster the whole house, but the painter can start painting with a 2 week lead after the plastering starts. This is because the plasterer will finish one part of the house first before moving on to other walls.

## Activity Sequencing

- So we can't just do activities in any old order. We need to figure out their dependencies (if any).
- We figure out dependencies as a project team. All team members are involved in thinking through what needs to be done first.

## Network Diagrams

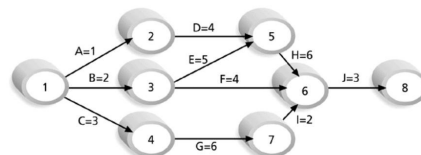
- We then need to represent these dependencies.
- One way is with Network Diagrams.



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

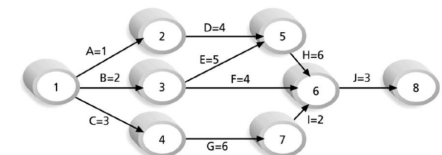
## Activity on Arrow Diagrams

- Here in an AOA diagram, the activities are labelled A through J.
- A, B, and C can all begin together.
- A takes 1 day, B takes 2 days and C takes 3 days to complete.
- E and F are dependent on B.



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

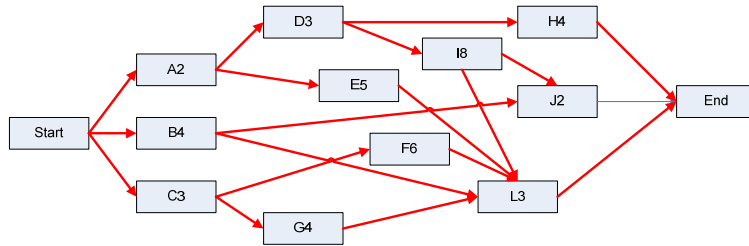
- They are called Activity on Arrow because the lines describe the tasks. The “nodes” or circles are just for reference.



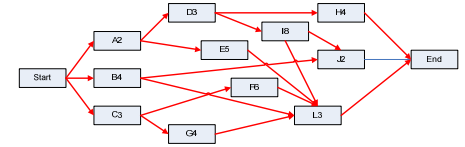
Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

# Activity on Node Diagrams

- In an AON Diagram, the nodes (usually shown as squares) are the activities. The lines/arrows are dependencies.



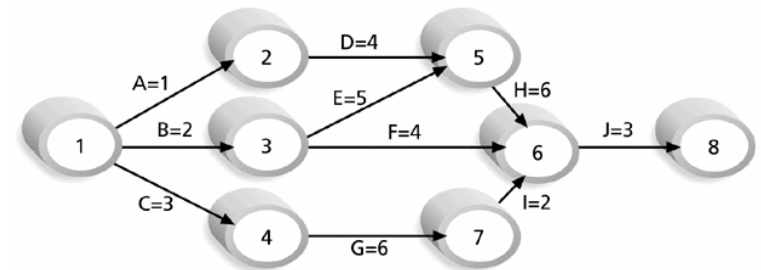
- A is an activity that takes 2 days, B takes 4 days and C takes 3 days.
- L takes 3 days but it is dependent on I, E, F, B, and G finishing before it can start.



# Critical Path

- The **Critical Path** is the shortest amount of time we can complete the project.
- We determine Critical Path by making a left-to-right pass across the network and finding the **earliest start time** each task can start. By the time we get to the end task, we know the earliest time the whole project can be finished.

# Critical Path Example



Note: Assume all durations are in days.

Path 1: A-D-H-J Length = 1+4+6+3 = 14 days  
 Path 2: B-E-H-J Length = 2+5+6+3 = 16 days  
 Path 3: B-F-J Length = 2+4+3 = 9 days  
 Path 4: C-G-I-J Length = 3+6+2+3 = 14 days

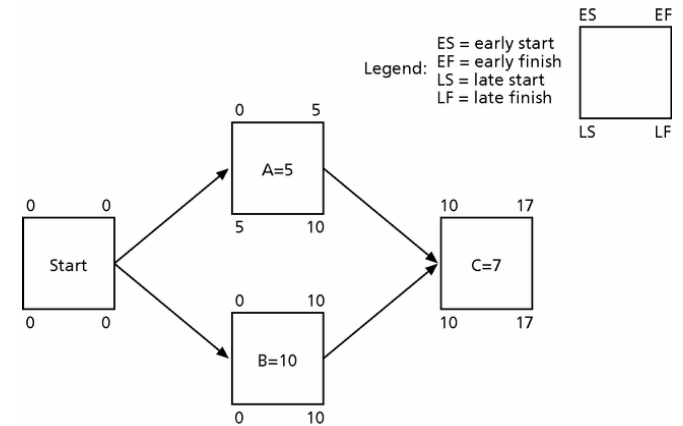
Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.



## Slack

- **Slack** is the amount of time a task can be delayed without changing the Critical Path.
- A task on the Critical Path has no (zero) slack.
- We calculate Slack by making a right-to-left pass across the network and find the **latest start date** for each task.

- Latest Start Date - Earliest Start Date = Slack
- To find Slack:



## References

- Meredith, Jack R., Samuel J. Mantel Jr. *Project Management: A Managerial Approach*. John Wiley & Sons, Inc. New York. 2003.
- Project Management Institute. *A Guide to the Project Management Body of Knowledge*. Third Edition. PMI. Newtown Square, PA. 2005.