SWINBURNE UNIVERSITY OF TECHNOLOGY

SWE20001 Managing Software Projects

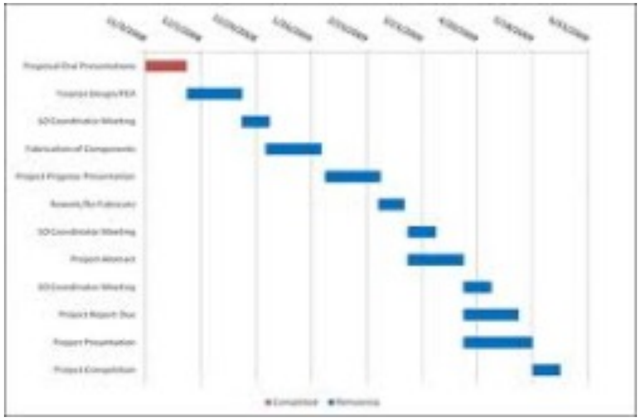
Lecture 9a

Traditional Task Scheduling

Project Scheduling [Trad. PM approach]

Goal:

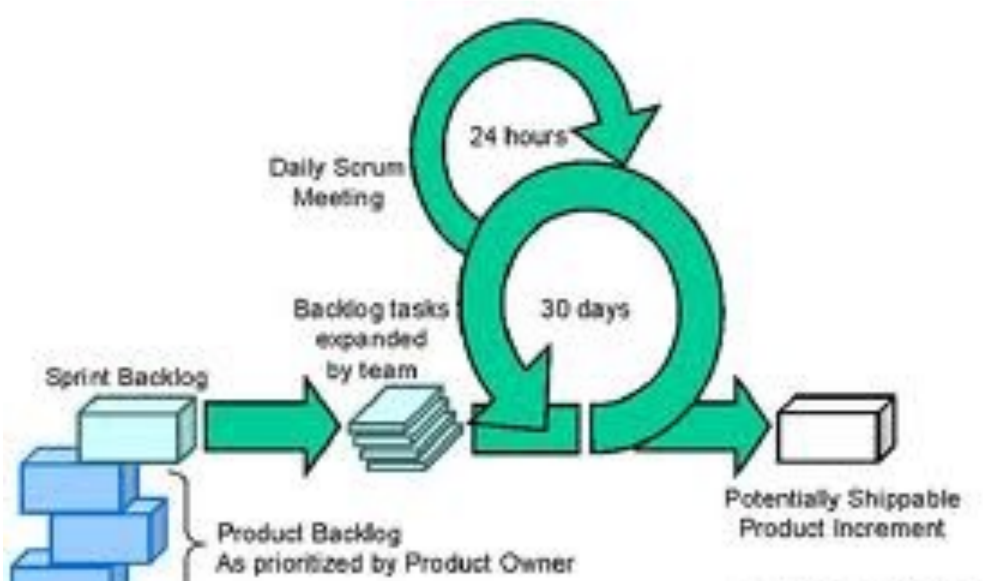
- Organizing project activities in a coherent order



- Organizing the staff allocations to activities to ensure project is completed in minimum time

Project Scheduling and Agile

- Traditional scheduling does not make much sense in agile projects
- Can use traditional approaches for sprint planning
- Still important to look at traditional approaches, as they are still used in industry, especially in non-development projects


24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
30 days
Sprint Backlog
Product Backlog As prioritized by Product Owner
Potentially Shippable Product Increment

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle

Traditional vs Agile

Predictive scheduling (“traditional”)

- Scope is known upfront and does not change (a lot)
- Problem/solution space is very well understood
- A detailed plan for most of the project activities can be done upfront
- Activities, activity dependencies and risks calculated
- Assume (at least in the first instance) that staff will be available whenever an activity is due to start

Adaptive scheduling (“agile”)

- Project defined mainly in terms of goals and vision, but scope may change (substantially)
- Problem/solution space not clearly defined
- Detailed plan can only be created for a short term (3-4 weeks)
- Risks are managed “dynamically”

Traditional vs Agile (cont’d)

Activity based (“traditional)

- Fix amount of work to be completed
- Define schedule and allocate resources so that all tasks can be completed

☞ *Mainly used in predictive planning and scheduling*

Time-boxed (“agile”)

- Fix amount of time available
- Choose scope so that it can be completed within time-box

☞ *Basic “building block” of most agile methodologies*

☞ *Sprint level – each sprint is time-boxed*

☞ *Task level – it is still Activity based!*

Trad. Example – Activities and Dependencies

Project: To replace a legacy system – very high level WBS

Act. Id.	Activity Name	Dependencies
A	Hardware Selection and Delivery	
B	Hardware Installation	A
C	Legacy System Analysis	
D	Software Design	C
E	System Implementation and Testing	B,D
F	Data Migration	C
G	System Deployment	E,F

Here
“X depends on Y” means that Y must be completed before X starts

Agile Example – Tasks and Dependencies

Item: To produce a sales report – very low level task breakdown

Task No.	Task Description	Dependencies
A	Select appropriate database server (may need spiking)	
B	Prepare database tables for coding	A
C	Analyze the reporting requirements	
D	Design the required module	C
E	Programming the required module	B,D
F	Design test data for reporting	C
G	Verify the correctness of report	E,F

Here
"X depends on Y" means that Y must be completed before X starts

Agile Example – Tasks, Dependencies and Duration

Task Id.	Task Description	Dependencies	Duration (hours)
A	Select appropriate database server (may need spiking)		4
B	Install database server for coding	A	3
C	Analyze the reporting requirements		5
D	Design the required module	C	4
E	Programming the required module	B,D	5
F	Design test data for reporting	C	3
G	Verify the correctness of report	E,F	2

Total number of hours = 26 hrs → 3.25 days work [Ideal time]
OR → 4.6 days work [realistic time]

Realistic = ideal / 0.7 (assume 70% of efficiency)

Agile Example – Tasks, Dependencies and Duration (cont)

Sequential technique

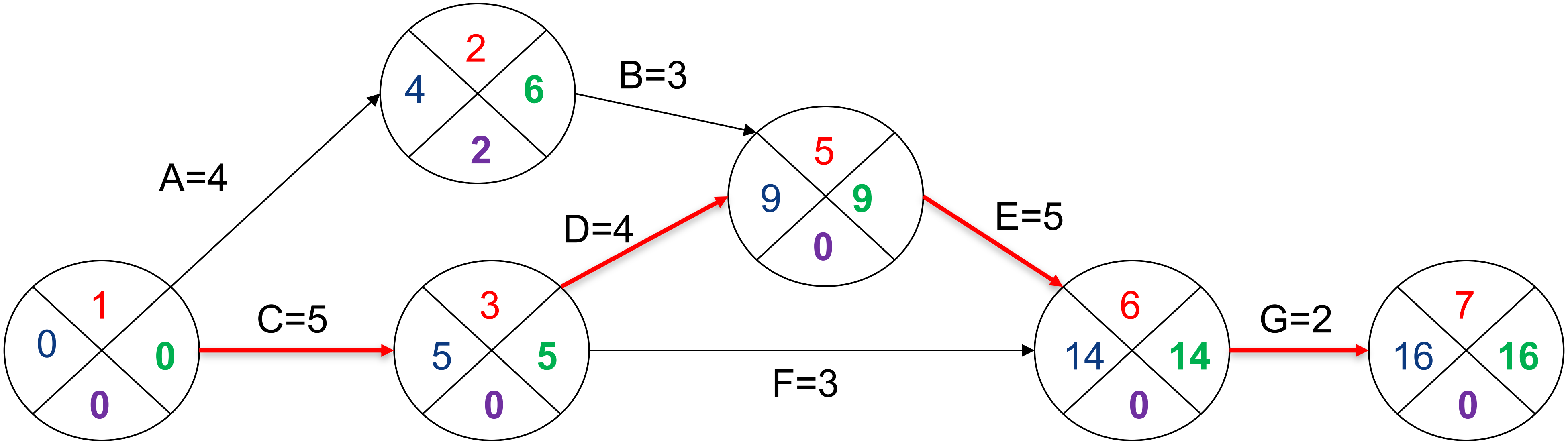
- Total time = 26 hours
- → 3.25 days [Ideal]
- → 4.6 days [Realistic]

Critical Path Method

- Total time = 16 hours
- → 2 days [Ideal]
- → 2.8 days [Realistic]

See next slide for a CPM diagram

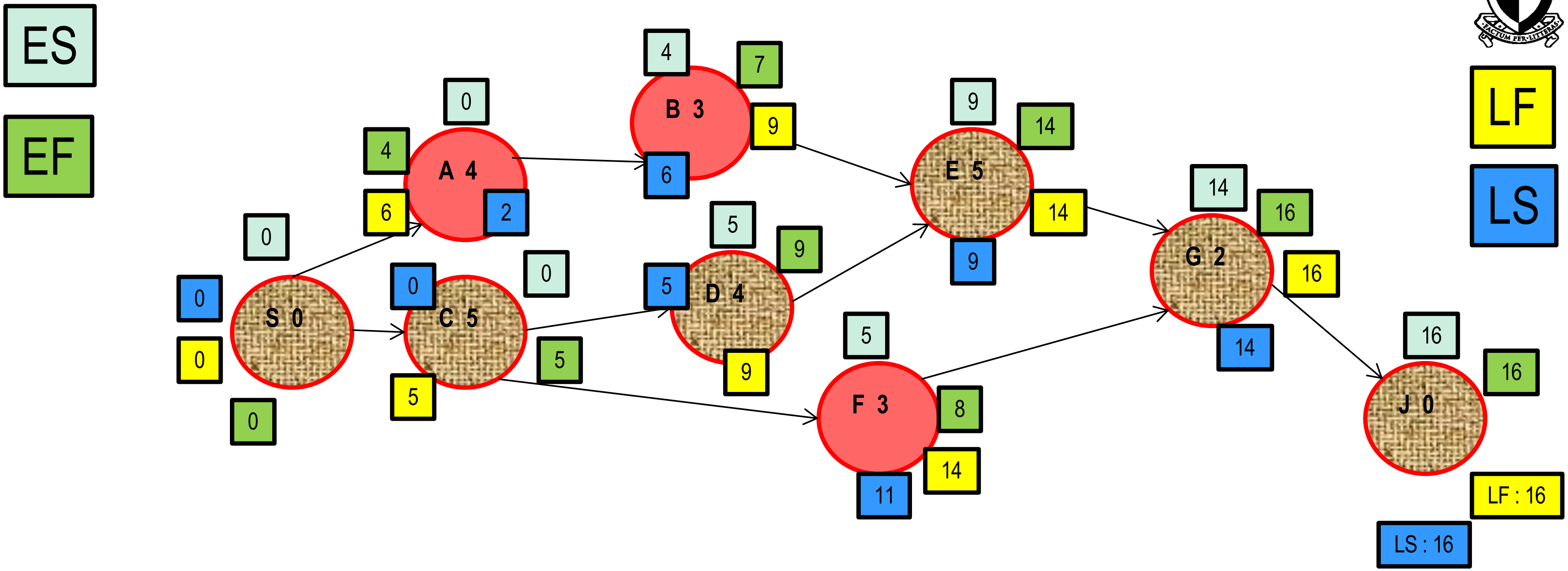
Example – CPM Diagram (Activities on Arrows)



Critical path links those nodes with no slack via the links (activities) that force there to be no slack.

The CP is activities C,D,E,G

Example – CPM Diagram (Activities on Nodes)



The critical path contains the nodes where the ES = LS (and EF = LF) – marked in diagram. We can also work out the slack and float figures for each activity (LS – ES).

By the way, there is no guarantee of a SINGLE critical path; there may be several!

Example – CPM Diagram (my own drawing)