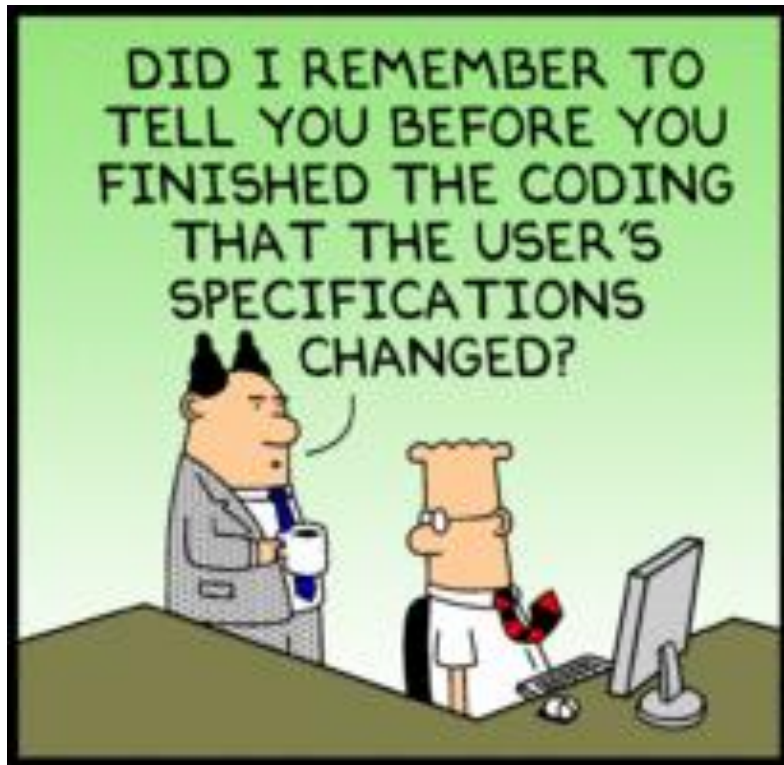


Scope Creep



Dilbert.com DilbertCartoonist@gmail.com



5-16-11 © 2011 Scott Adams, Inc. Dist. by Universal Uclick.



Time Management



CS352 Project Management for Computer Scientists

Scope and Time Management



interact at:

warwick.ac.uk/pm4cs/2

Dr. Ian Saunders

Recap: Project Selection Seminar

- Outputs / Outcomes / Benefits
- SMART Objectives
 - **S**pecific, **M**easurable, **A**chievable, **R**elevant, **T**ime-bound (SMART)
- Present a convincing Business Case
- **Assigned a Case Study**

Seminar issues?

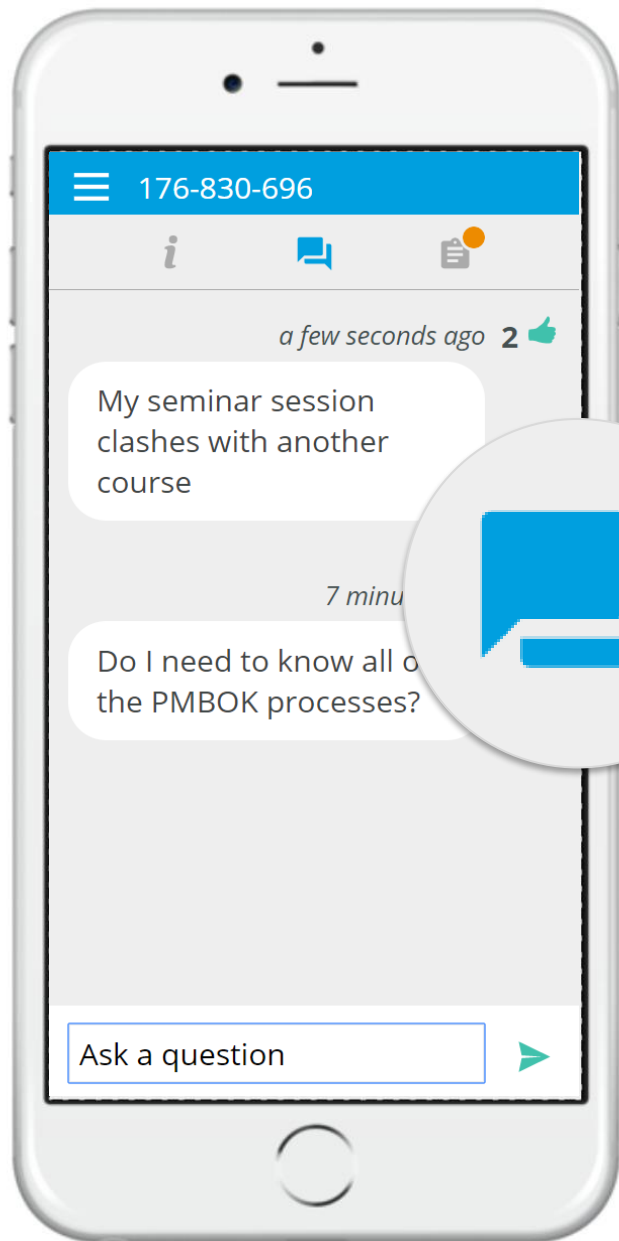


warwick.ac.uk/pm4cs/2

Any seminar issues?

Vote for up to 3 choices

1. No issues
2. I missed the first seminar
3. I'm not in a seminar team / group
4. I have a timetable clash



Ask a question at any time!
warwick.ac.uk/pm4cs/2

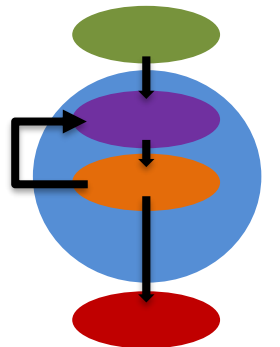
Recap: Week 1



- A project is a temporary endeavour undertaken to create a unique product service or result.
- Manage the *triple constraint*: cost, schedule, and scope
- But also quality, risk, resources, stakeholders...



- PMBOK® Guide – 5PGs, 10 KAs, 49 Processes
- Initiation:
 - Project Mandate – *project created*
 - Project Charter – *PM assigned and given authority to begin...*



Quiz!



warwick.ac.uk/pm4cs/2

One of the following groups is NOT a process group
– but which one?

1. Planning
2. Integrating
3. Closing
4. Executing

Which is the knowledge area where requirements are collected?

1. Communication
2. Scope
3. Integration
4. Quality

Which process group has the greatest number of processes?

1. Monitoring and Controlling
2. Planning
3. Executing
4. Integrating



Which process group up most of the budget in a project?

1. Initiating
2. Planning
3. Executing
4. Monitoring and Controlling
5. Closing



Leaderboard



Position	Participants	Score

Which of the following is NOT one of the three initiating process outputs?

1. Business Case
2. Stakeholder management strategy
3. Stakeholder register
4. Project Charter

You communicate with the development team in

1. Integration which knowledge area?

2. Stakeholder

3. Resources

4. Communication

In which process group will you perform team building?

1. Initiating
2. Planning
3. Executing
4. Monitoring and Controlling

The Monitoring and Controlling process group contains an output called Accepted Deliverables which process is this output from?

1. Perform Quality Assurance
2. Verify Scope
3. Control Scope
4. Perform Quality Control



Leaderboard



Position	Participants	Score

Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Objectives

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Objectives

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

S
M
A
R
T

Objectives

By the end of this lecture you should have learned:

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

S
M
A
R
T ✓

Objectives

By the end of this lecture you should have learned:

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)

How to design • Work packages

Understand • Deliverable-oriented (vs Objective- / Process- oriented)

- Time Management

Four • Estimation techniques

How to show • Dependencies ~~and~~ using Gantt Charts

How to draw • Project Network Diagram

How to apply • Critical Path Method (CPM)

How to apply • Program Evaluation and Review Technique (PERT)

S ✓
M
A
R
T ✓

Objectives

By the end of this lecture you should have learned:

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)

How to design Work packages

Understand Deliverable-oriented (vs Objective- / Process- oriented)

- Time Management

Four Estimation techniques

How to show Dependencies and Gantt Charts

How to draw a Project Network Diagram

How to apply Critical Path Method (CPM)

How to apply Program Evaluation and Review Technique (PERT)

Exam →

S	✓
M	✓
A	
R	
T	✓

Objectives

By the end of this lecture you should have learned:

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)

How to design Work packages

Understand Deliverable-oriented (vs Objective- / Process- oriented)

- Time Management

Four Estimation techniques

How to show Dependencies and Gantt Charts

How to draw a Project Network Diagram

How to apply Critical Path Method (CPM)

How to apply Program Evaluation and Review Technique (PERT)

	S	✓
Exam →	M	✓
Experience →	A	✓
	R	
	T	✓

Objectives

By the end of this lecture you should have learned:

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)

How to design • Work packages

Understand • Deliverable-oriented (vs Objective- / Process- oriented)

- Time Management

Four • Estimation techniques

How to show • Dependencies and Gantt Charts

How to draw • Project Network Diagram

How to apply • Critical Path Method (CPM)

How to apply • Program Evaluation and Review Technique (PERT)

	S	✓
Exam →	M	✓
Experience →	A	✓
Course objectives →	R	✓
	T	✓

Today

- **Planning is a continuous process**
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Developing a Project Plan

Once the project is given the go-ahead and the Charter is signed-off by the Sponsor, the Project Manager takes over.

First steps:

1. Breakdown work, Plan Timeline
2. Set out milestones, identify bottlenecks
3. Plan Budget, Allocate Resources

Developing a Project Plan

Once the project is given the go-ahead and the Charter is signed-off by the Sponsor, the Project Manager takes over.

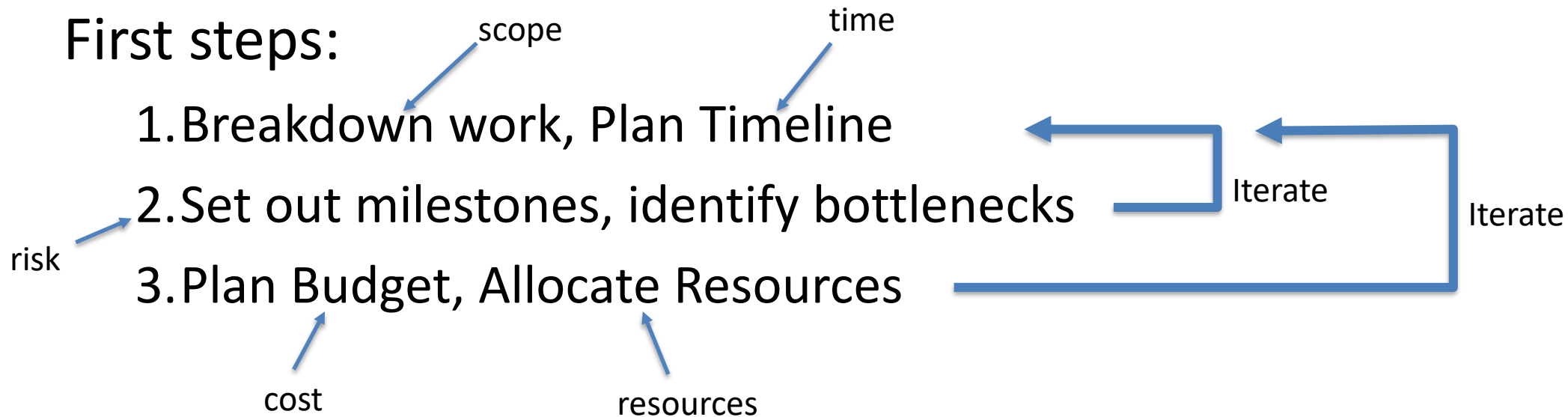
First steps:

-
- The diagram illustrates the relationship between project constraints and the first steps of developing a project plan. It features a list of three steps, with blue arrows pointing from constraint labels to specific steps: 'scope' points to step 1, 'time' points to step 1, 'risk' points to step 2, 'cost' points to step 3, and 'resources' points to step 3.
1. Breakdown work, Plan Timeline
 2. Set out milestones, identify bottlenecks
 3. Plan Budget, Allocate Resources

Developing a Project Plan

Once the project is given the go-ahead and the Charter is signed-off by the Sponsor, the Project Manager takes over.

First steps:



	Initiating	Planning	Executing	Monitor/Control	Closing
Integration	Develop Project Charter	Develop Project Management Plan	Direct and Manage Project Work Manage Project Knowledge	Monitor and Control Project Work Perform Integrated Change Control	Close Project or Phase
Scope		Plan Scope Management Collect Requirements Define Scope Create WBS		Validate Scope Control Scope	
Time		Plan Schedule Management Define Activities Sequence Activities Estimate Activity Durations Develop Schedule		Control Schedule	
Cost		Plan Cost Management Estimate Costs Determine Budget		Control Costs	
Quality		Plan Quality Management	Manage Quality	Control Quality	
HR/Resources		Plan Resource Management Estimate Activity Resources	Acquired Resources Develop Team Manage Team	Control Resources	
Communication		Plan Communications Management	Manage Communications	Monitor Communications	
Risk		Plan Risk Management Identify Risks Perform Qualitative Risk Analysis Perform Quantitative Risk Analysis Plan Risk Responses	Implement Risk Responses	Monitor Risks	
Procurement		Plan Procurement Management	Conduct Procurements	Control Procurements	Close Procurements
Stakeholder	Identify Stakeholders	Plan Stakeholder Engagement	Manage Stakeholder Engagement	Monitor Stakeholder Engagement	



Planning Stage

	Planning
Integration	Develop Project Management Plan
Scope	Plan Scope Management Collect Requirements Define Scope Create WBS
Time	Plan Schedule Management Define Activities Sequence Activities Estimate Activity Durations Develop Schedule
Cost	Plan Cost Management Estimate Costs Determine Budget
Quality	Plan Quality Management
HR/Resources	Plan Resource Management Estimate Activity Resources
Communication	Plan Communications Management
Risk	Plan Risk Management Identify Risks Perform Qualitative Risk Analysis Perform Quantitative Risk Analysis Plan Risk Responses
Procurement	Plan Procurement Management
Stakeholder	Plan Stakeholder Engagement

Planning Stage

Project **Scope** Management

Plan Scope Management

- How will *scope* be defined, validated and controlled?

Collect Requirements

- Document stakeholder requirements

Define Scope

- Project and Product Descriptions

Create WBS

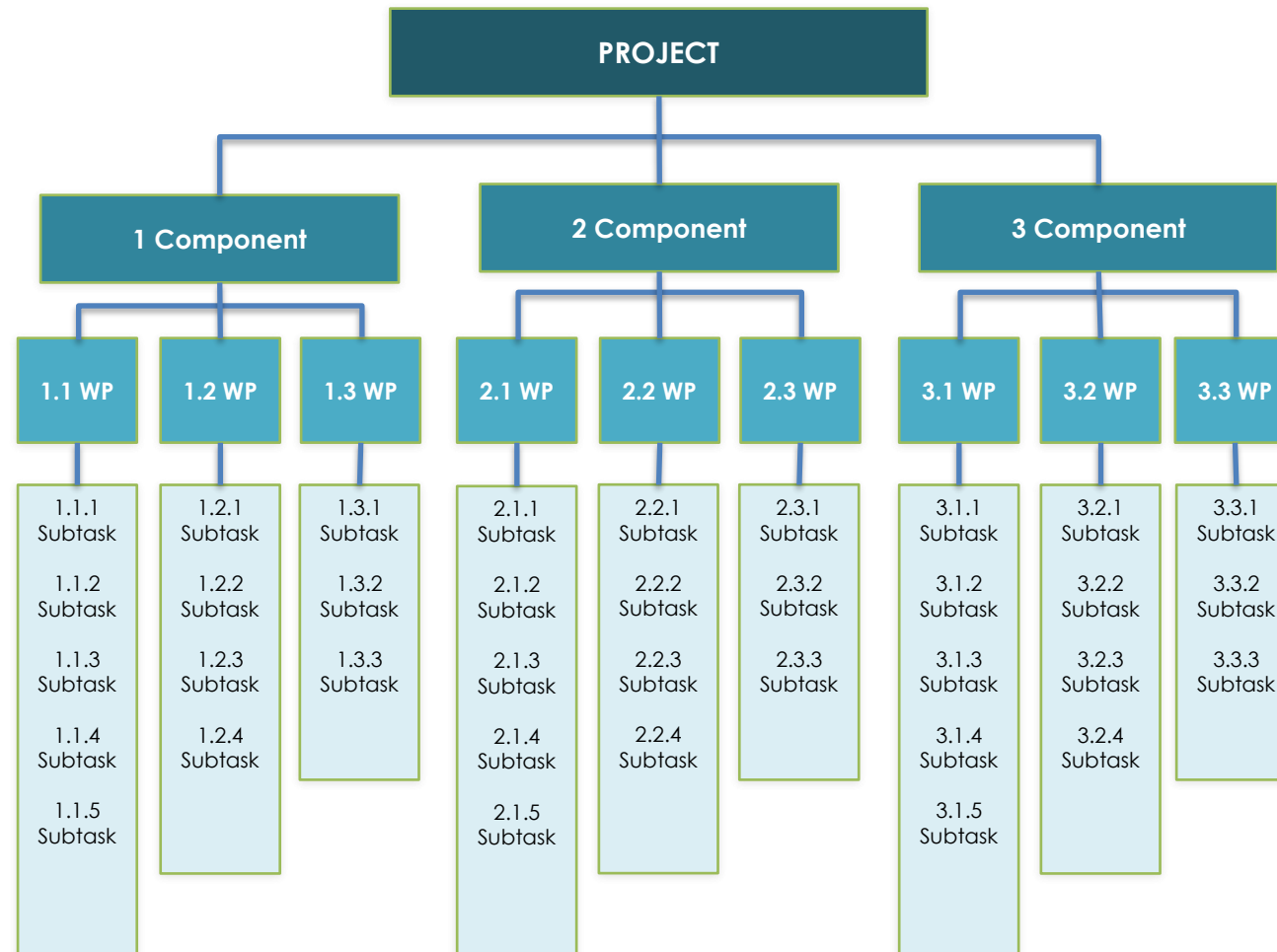
- Subdivide deliverables into smaller, more manageable components.

	Planning
Integration	Develop Project Management Plan
Scope	Plan Scope Management Collect Requirements Define Scope Create WBS
Time	Plan Schedule Management Define Activities Sequence Activities Estimate Activity Durations Develop Schedule
Cost	Plan Cost Management Estimate Costs Determine Budget
Quality	Plan Quality Management
HR/Resources	Plan Resource Management Estimate Activity Resources
Communication	Plan Communications Management
Risk	Plan Risk Management Identify Risks Perform Qualitative Risk Analysis Perform Quantitative Risk Analysis Plan Risk Responses
Procurement	Plan Procurement Management
Stakeholder	Plan Stakeholder Engagement

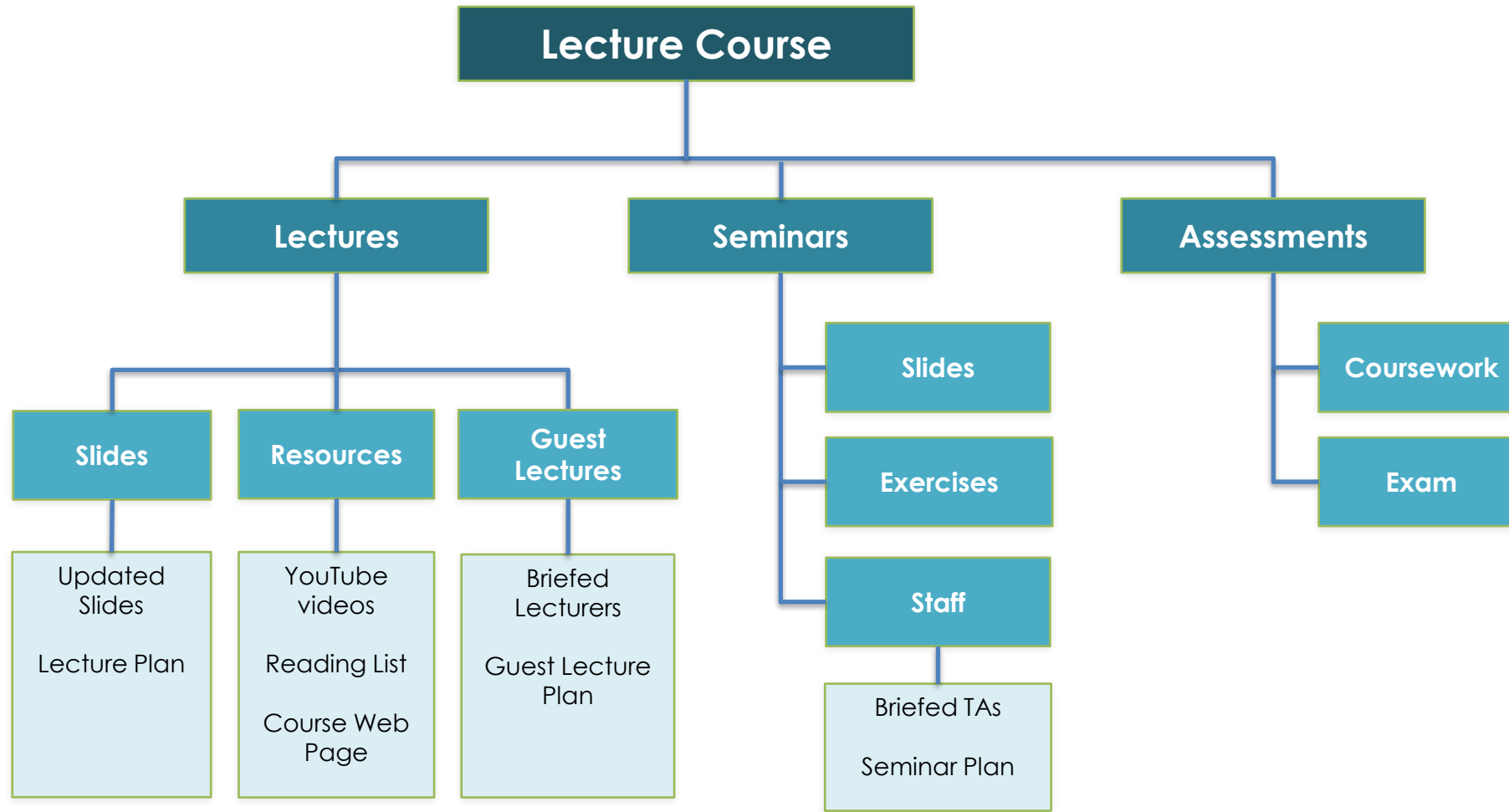
Today

- Planning is a continuous process
- **Scope Management: Work Breakdown Structure (WBS)**
 - **Work packages**
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Work Breakdown Structure



Work Breakdown Structure Example



Work Breakdown Structure

“A **deliverable-oriented hierarchical decomposition** of the **work** to be executed by the project team to accomplish the project **objectives** and create the required **deliverables**.

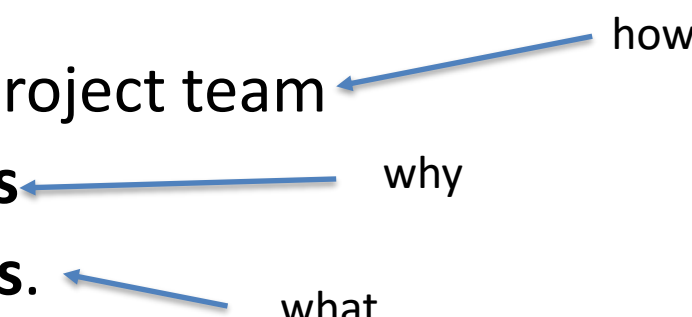
It organizes and defines the total **scope** of the project.

Each descending level represents an increasingly detailed definition of the project work... .”

– **PMBOK® Guide.**

Work Breakdown Structure

“A deliverable-oriented hierarchical decomposition of the **work** to be executed by the project team to accomplish the project **objectives** and create the required **deliverables**.”



how
why
what

It organizes and defines the total **scope** of the project.

Each descending level represents an increasingly detailed definition of the project work... .”

– PMBOK® Guide.

WBS - Definitions

Deliverables – Outputs. Must align with initial objectives.

WBS - Definitions

Deliverables - Outputs. Must align with initial objectives.

Work Package – Smallest unit of the WBS

- Set of related tasks and deliverables
- Clearly defined interactions with other WPs
- Clear identification of inputs, outputs, internal activities.
- Clear estimation of cost, duration, resources.

WBS - Definitions

Deliverables - Outputs. Must align with initial objectives.

Work Package – Smallest unit of the WBS

- Set of related tasks and deliverables
- Clearly defined interactions with other WPs
- Clear identification of inputs, outputs, internal activities.
- Clear estimation of cost, duration, resources.

Milestones - Points of control that separate Work Packages.

Parallelism

Risk of overrun

More WP interactions

Harder to define

Efficient use of team

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros		
Cons		

Work Packages

Risk of overrun

More WP interactions

Harder to define

Efficient use of team

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

	Small Work Package	Large Work Package
Pros	Parallelism	
Cons		

Work Packages

More WP interactions

Harder to define

Efficient use of team

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

	Small Work Package	Large Work Package
Pros	Parallelism	
Cons		Risk of overrun

Work Packages

Harder to define

Efficient use of team

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

	Small Work Package	Large Work Package
Pros	Parallelism	
Cons	More WP interactions	Risk of overrun

Work Packages

Efficient use of team

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

	Small Work Package	Large Work Package
Pros	Parallelism	
Cons	More WP interactions Harder to define	Risk of overrun

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism	Efficient use of team
Cons	More WP interactions Harder to define	Risk of overrun

Easier to manage team

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism	Efficient use of team Easier to manage team
Cons	More WP interactions Harder to define	Risk of overrun

Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism	Efficient use of team Easier to manage team
Cons	More WP interactions Harder to define	Risk of overrun Problems detected late

Estimable

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism Estimable	Efficient use of team Easier to manage team
Cons	More WP interactions Harder to define	Risk of overrun Problems detected late

Team knows best

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism Estimable	Efficient use of team Easier to manage team Team knows best
Cons	More WP interactions Harder to define	Risk of overrun Problems detected late

Easier to control team

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism Estimable Easier to control team	Efficient use of team Easier to manage team Team knows best
Cons	More WP interactions Harder to define	Risk of overrun Problems detected late

PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	Parallelism Estimable Easier to control team	Efficient use of team Easier to manage team Team knows best
Cons	More WP interactions Harder to define	Risk of overrun Problems detected late PM has less control

Duplication of work
between WPs

Work Packages

	Small Work Package	Large Work Package
Pros	<ul style="list-style-type: none">ParallelismEstimableEasier to control team	<ul style="list-style-type: none">Efficient use of teamEasier to manage teamTeam knows best
Cons	<ul style="list-style-type: none">More WP interactionsHarder to defineDuplication of work between WPs	<ul style="list-style-type: none">Risk of overrunProblems detected latePM has less control



Leaderboard



Position	Participants	Score

Total Participants: 0

Average Score: 0

WBS - Definitions

Deliverables - Outputs. Must align with initial objectives.

Work Package – Smallest unit of the WBS

- Set of related tasks and deliverables
- Clearly defined interactions with other WPs
- Clear identification of inputs, outputs, internal activities.
- Clear estimation of cost, duration, resources.

Milestones - Points of control that separate Work Packages.

Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - **Deliverable-oriented (vs Objective- / Process- oriented)**
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Work Breakdown Structure



Work Breakdown Structure

Good / Bad?



Work Breakdown Structure

These are Phases,
Not Deliverables!

Activities are
not Deliverables!



Work Breakdown Structure

Not good as WPs!
(can't easily be parallelised)



This is waterfall!

Deliverables vs Objectives

- Objectives:
 - Desired **benefits, outcomes, or improvements**
 - *“To [improve/increase/enhance] something by [amount] by [date]”*
 - **Specific, Measurable, Achievable, Relevant, Time-bound (SMART)**
- Deliverables:
 - **Outputs or products**
 - stuff we will deliver to the customer
 - *Produced to achieve objectives*

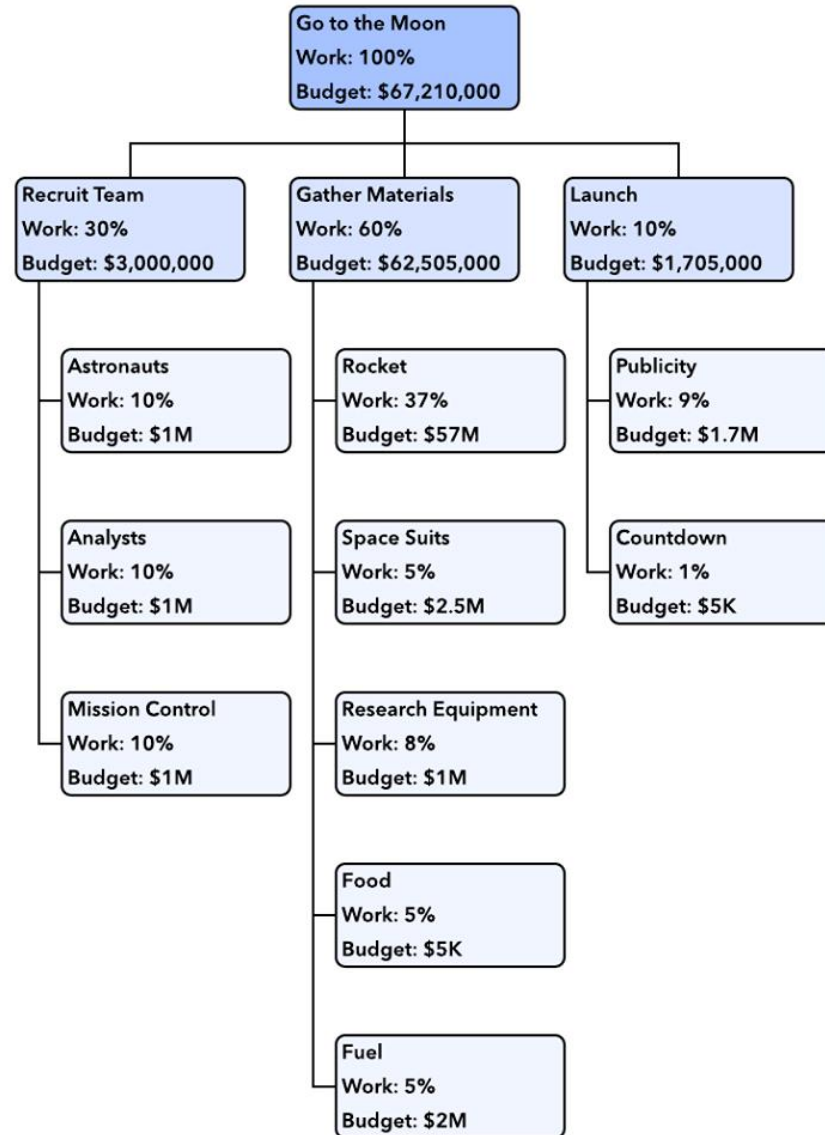
Deliverables vs Objectives

- Objectives:
 - Relate to **purpose** – *why are we going to build it?*
- Deliverables:
 - Relate to **work** – *what are we going to build?*
 - Define **scope**
 - **Estimable, Parallelisable, Purchasable**

Deliverables vs Objectives

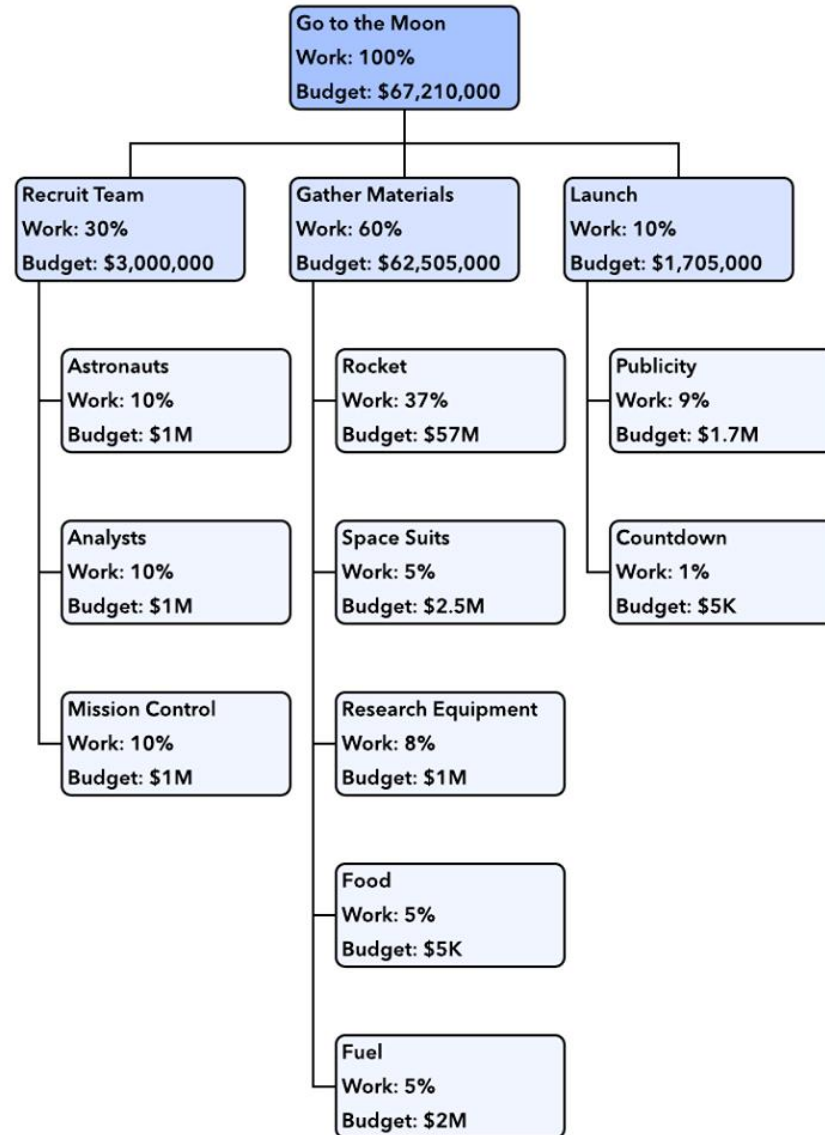
- **Both important for scope management:**
 - Must align outputs with customer's objectives
 - Must exclude deliverables from your project scope that do not help to achieve the project objectives
 - (Avoid *scope creep*...)

Work Breakdown Structure

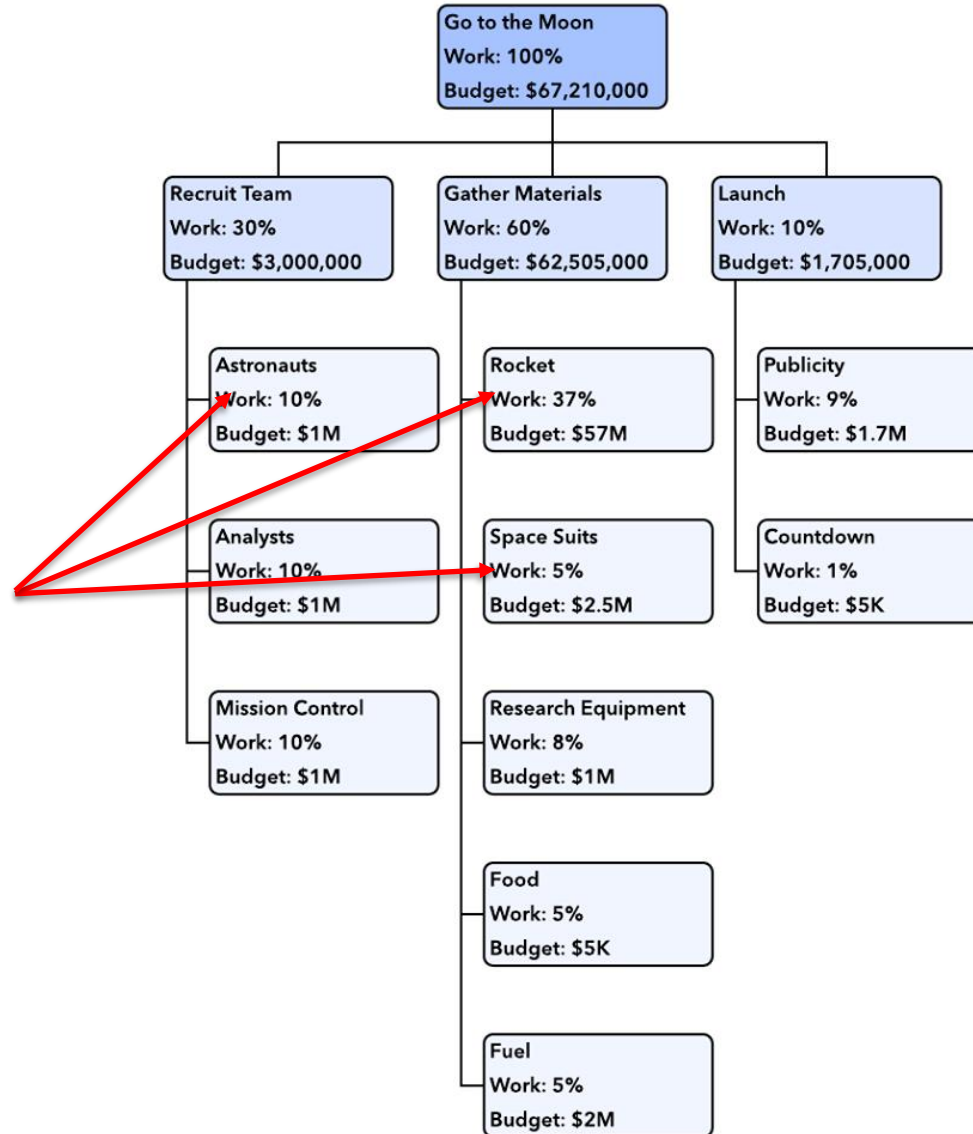


Work Breakdown Structure

Good / Bad?



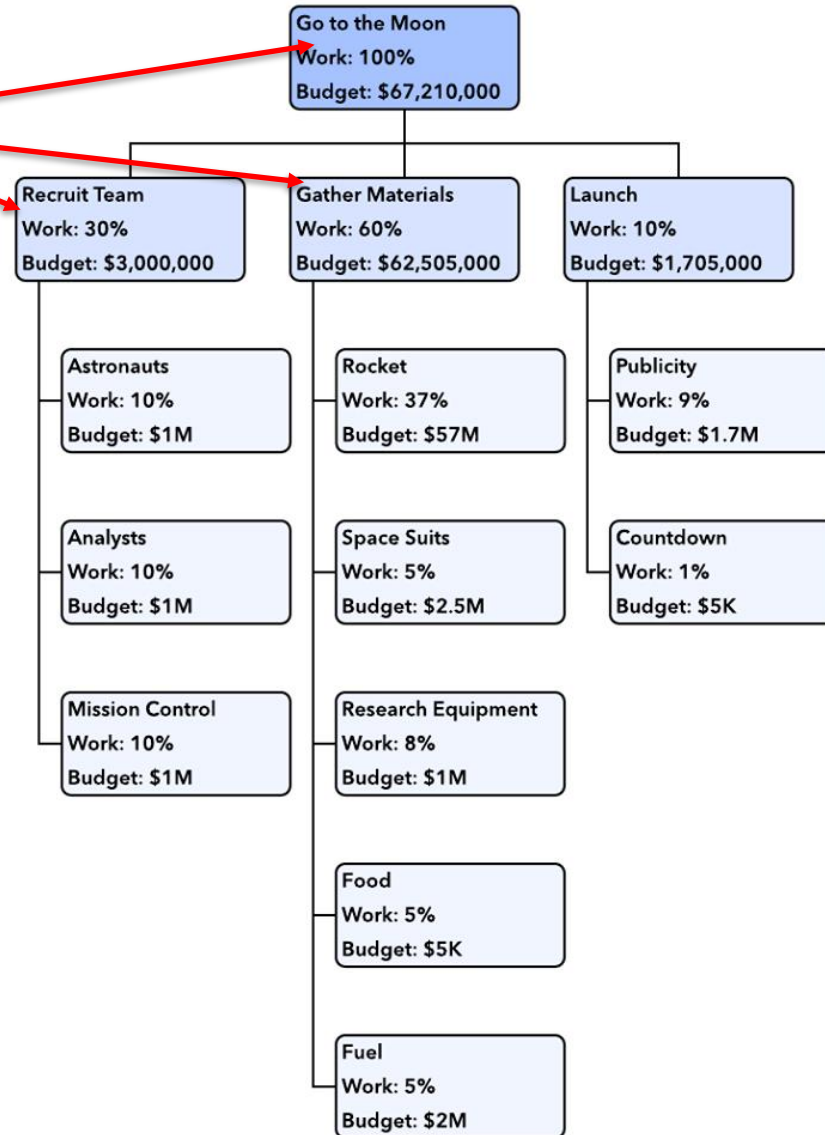
Work Breakdown Structure



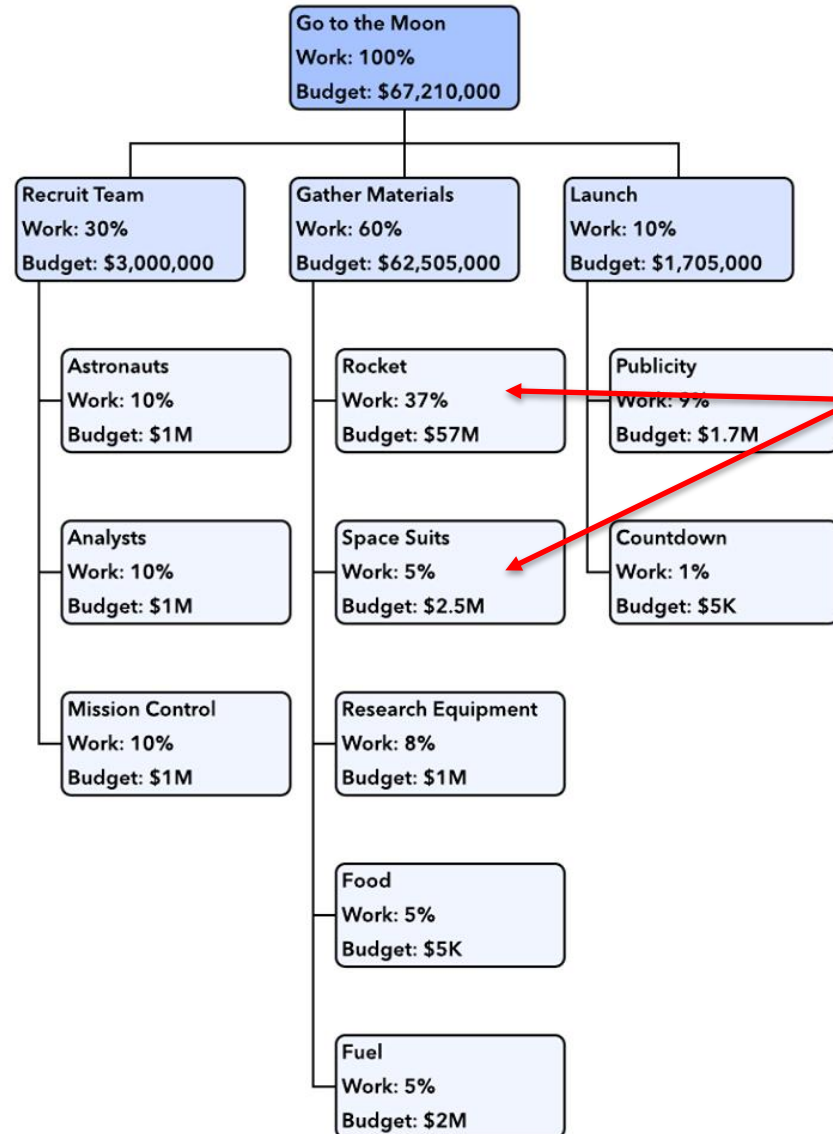
Deliverables!
(**scope** is clear)

Work Breakdown Structure

Objectives
(**purpose** is clear)



Work Breakdown Structure



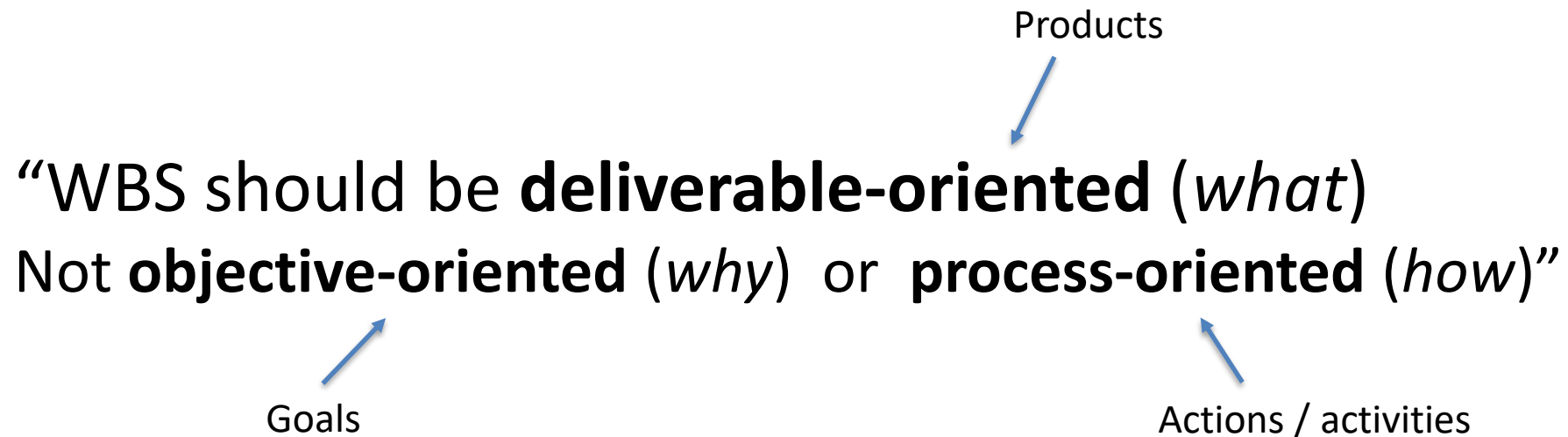
Each WP is well defined:

- Unambiguous
- Understandable
- Estimable
- Parallelisable
- Purchasable

Deliverable-Oriented WBS

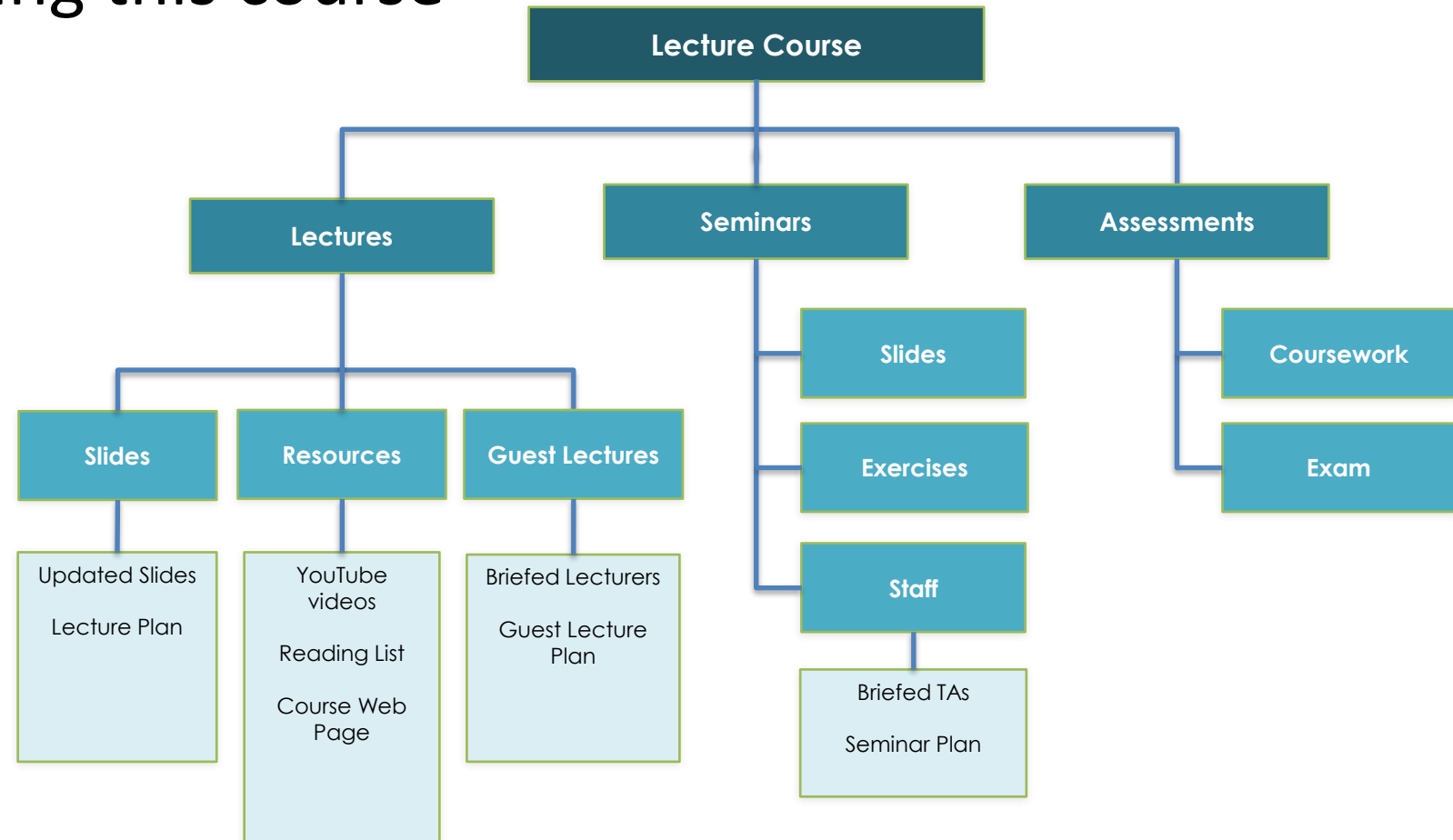
“WBS should be **deliverable-oriented** (*what*)
Not **objective-oriented** (*why*) or **process-oriented** (*how*)”

Deliverable-Oriented WBS



WBS Example: What is the deliverable?

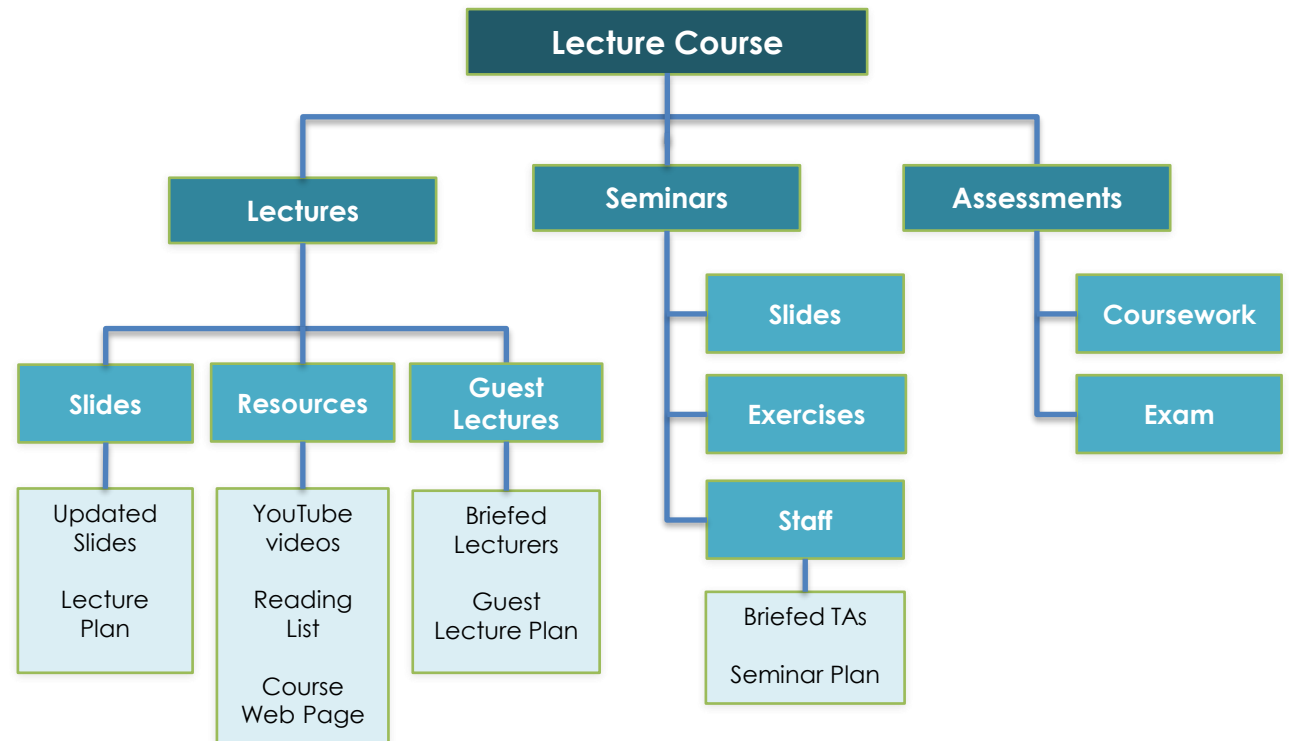
- Project: Teaching this course



WBS Example: What is the deliverable?

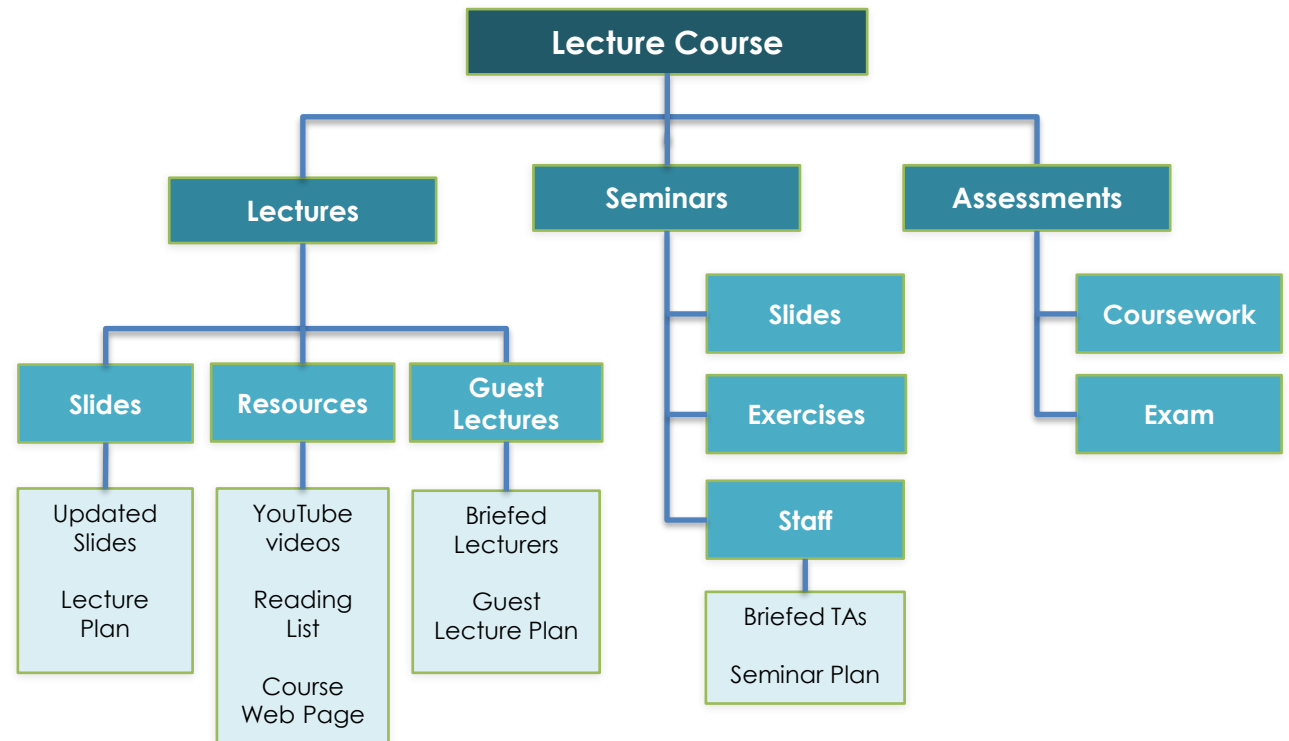
- Project: Teaching this course
- Objective: To increase student's understanding of lecture material.

Is the objective SMART enough?
We may need to break it down further...



WBS Example: What is the deliverable?

- Project: Teaching this course
- Objective: To increase student's understanding of some very boring lecture material.



WBS Example: What is the deliverable?

- Project: Teaching this course
- Objective: To increase student's understanding of *some very boring* lecture material.
 - Problem: Students can't pay attention for a full two hours
 - Solution: Students engage more when there is 'active learning'

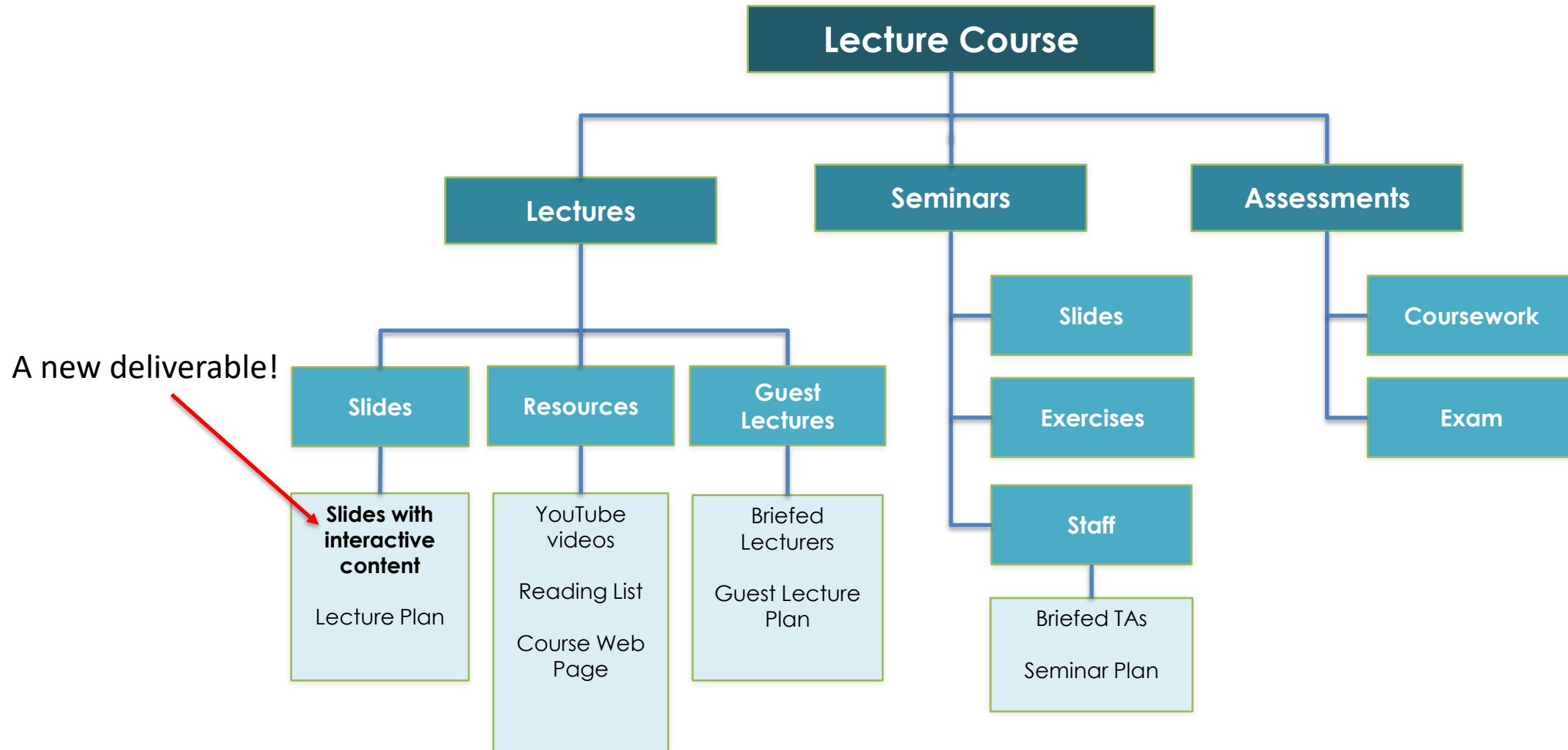


The creative bit: brainstorming potential solutions
(The project management textbook can't help you here...)

WBS Example: What is the deliverable?

- Project: Teaching this course
- Objective: To increase student's understanding of *some very boring* lecture material.
 - Problem: Students can't pay attention for a full two hours
 - Solution: Students engage more when there is 'active learning'
 - **Deliverable: interactive content**

WBS Example: What is the deliverable?



A WBS should be deliverable-oriented, not objective-oriented or process-oriented

1. Agree
2. Disagree

WBS Example: What is the deliverable?

- Project: Recognising emotions from faces (3rd year project)
- Objective: Discover a magical algorithm to solve it.
- ~~Deliverable: a magical algorithm~~



A bit too vague...

- How much time will it take?
- What resources are required?
- Does the team have the necessary skills?

WBS Example: What is the deliverable?

- Project: Recognising emotions from faces (3rd year project)
- Objective: Implement a magical Machine Learning solution.



A refined objective gives us a better place to start

WBS Example: What is the deliverable?

- Project: Recognising emotions from faces (3rd year project)
- Objective: Implement a magical Machine Learning solution.
 - Problem: The machine learning course is not until next term
 - Solution: Do some research, read papers.

WBS Example: What is the deliverable?


- Project: Recognising emotions from faces (3rd year project)
- Objective: Implement a magical Machine Learning solution.
 - Problem: The machine learning course is not until next term
 - Solution: Do some research, read papers.
 - Deliverable: List of candidate algorithms and their suitability



An answer to the research question is the deliverable.

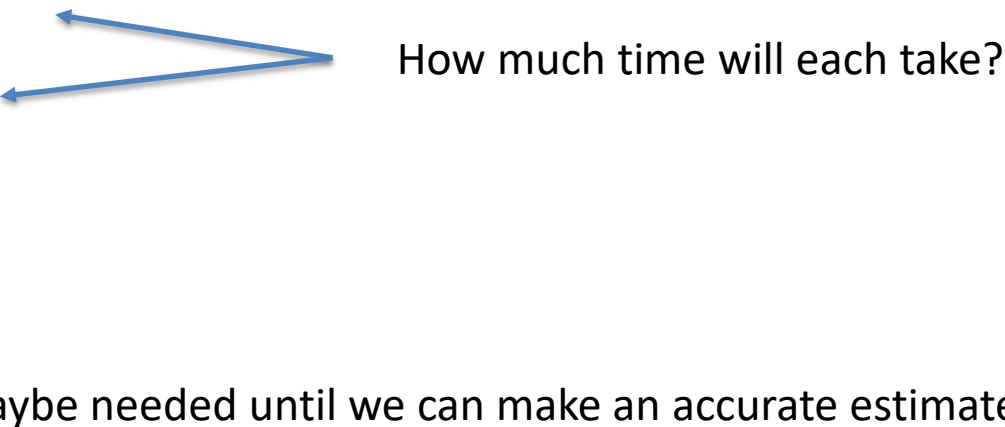
WBS Example: What is the deliverable?

- Project: Recognising emotions from faces (3rd year project)
- Objective: Implement a magical Machine Learning solution.
 - **New Problem:** These algorithms are very complex and require lots of training data in a very specific format.



Research is a high-risk activity:
it can change the scope!

WBS Example: What is the deliverable?

- Project: Recognising emotions from faces (3rd year project)
 - Objective: Implement a magical Machine Learning solution.
 - New Problem: These algorithms are very complex and require lots of training data in a very specific format.
 - Deliverables:
 - List of candidates and their suitability
 - Tagged training data
 - A working implementation of X
- 
- How much time will each take?
- Still too vague?
More research maybe needed until we can make an accurate estimate

Deliverable-Oriented WBS

“WBS should be **deliverable-oriented** (*what*)
Not **objective-oriented** (*why*) or **process-oriented** (*how*)”

Pros:

- Deliverables are easier to estimate than objectives
- Makes us focus on the essentials
- Once scope is defined, the ‘what’ is fixed, but ‘how’ remains fluid
- Allows us to manage **scope**

Deliverable-Oriented WBS

“WBS should be **deliverable-oriented** (*what*)
Not **objective-oriented** (*why*) or **process-oriented** (*how*)”

Pros:

- Deliverables are easier to estimate than objectives
- Makes us focus on the essentials
- Once scope is defined, the ‘what’ is fixed, but ‘how’ remains fluid
- Allows us to manage **scope**

Cons:

- Loses sight of the objectives
- Restricts further scope changes and creativity from developers
- May overlook costly processes that consume time and resources
- Doesn’t come naturally

Deliverable-Oriented WBS

The Problem:

- We want to stay focused on **objectives and goals**
 - But not present in the final WBS!
- We also want to use it plan the **schedule and budget**
 - It's actions and processes that take time and cost money, not the deliverable itself!
- We want to encourage **creativity**
 - We don't want to fix the scope!

Deliverable-Oriented WBS

The Problem:

- We want to stay focused on **objectives and goals**
 - But not present in the final WBS!
- We also want to use it plan the **schedule and budget**
 - It's actions and processes that take time and cost money, not the deliverable itself!
- We want to encourage **creativity**
 - We don't want to fix the scope!

One solution:



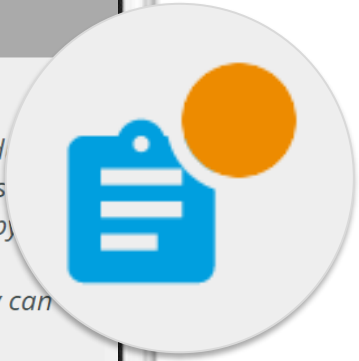
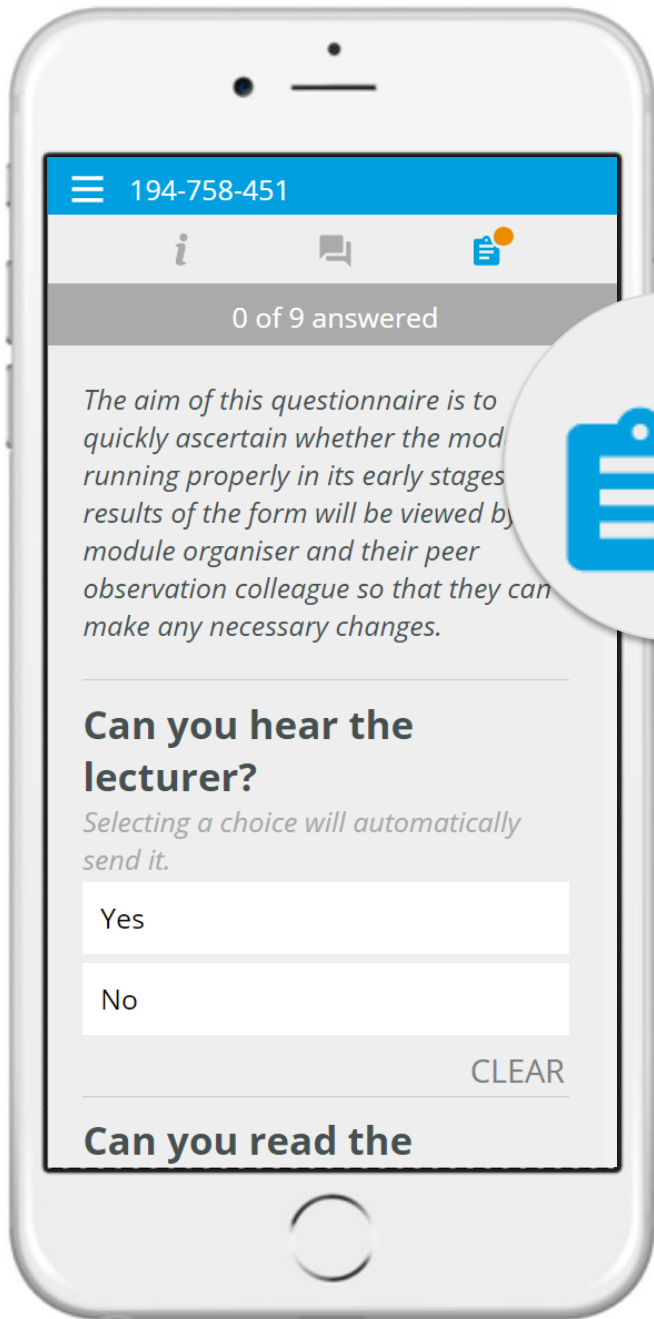
- We can (If we break the rules!)
- Forget deliverables, break it down **logically**, working backwards.
 - Fine as long as we get to the deliverables.



- We can!
- We will decide *how* we will do the work *after* we have decided *what*!



- Creativity is at the discretion of the PM
- WBS is a tool to fix the scope!



Please complete the
Initial Module Feedback Survey

warwick.ac.uk/pm4cs/2

Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- **Time Management**
 - **Estimation**
 - **Dependencies and Gantt Charts**
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Planning Stage

	Planning
Integration	Develop Project Management Plan
Scope	Plan Scope Management Collect Requirements Define Scope Create WBS
Time	Plan Schedule Management Define Activities Sequence Activities Estimate Activity Durations Develop Schedule
Cost	Plan Cost Management Estimate Costs Determine Budget
Quality	Plan Quality Management
HR/Resources	Plan Resource Management Estimate Activity Resources
Communication	Plan Communications Management
Risk	Plan Risk Management Identify Risks Perform Qualitative Risk Analysis Perform Quantitative Risk Analysis Plan Risk Responses
Procurement	Plan Procurement Management
Stakeholder	Plan Stakeholder Engagement

Planning Stage

Project **Time** Management

Plan Schedule Management

- How will *the schedule* be planned, managed, executed and controlled?

Define Activities

- Which *actions* will produce the *deliverables*?

Sequence Activities

- Relationships / dependencies

Estimate Activity Durations

- How long will each take?

Develop Schedule

- Balancing constraints to create a timeline.

	Planning
Integration	Develop Project Management Plan
Scope	Plan Scope Management Collect Requirements Define Scope Create WBS
Time	Plan Schedule Management Define Activities Sequence Activities Estimate Activity Durations Develop Schedule
Cost	Plan Cost Management Estimate Costs Determine Budget
Quality	Plan Quality Management
HR/Resources	Plan Resource Management Estimate Activity Resources
Communication	Plan Communications Management
Risk	Plan Risk Management Identify Risks Perform Qualitative Risk Analysis Perform Quantitative Risk Analysis Plan Risk Responses
Procurement	Plan Procurement Management
Stakeholder	Plan Stakeholder Engagement

Estimating Activity Durations

Analogous – How long it took last time (adjusted to this project)

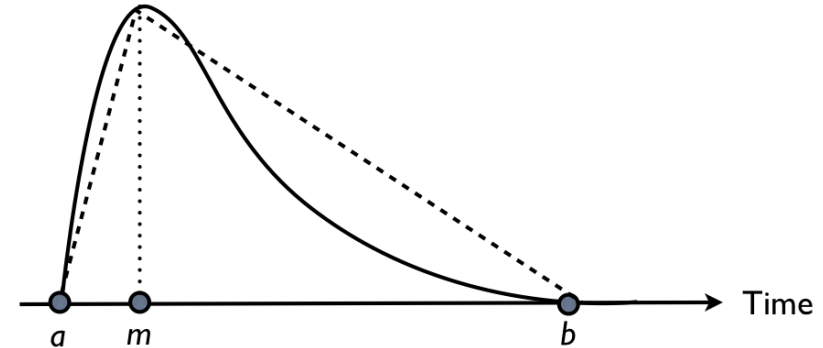
Parametric – As above, but with a statistical model

Team-based – by the people doing the work

Three Point – m mean, a min, b max

$$t_{\text{triangular}} = (a + m + b)/3$$

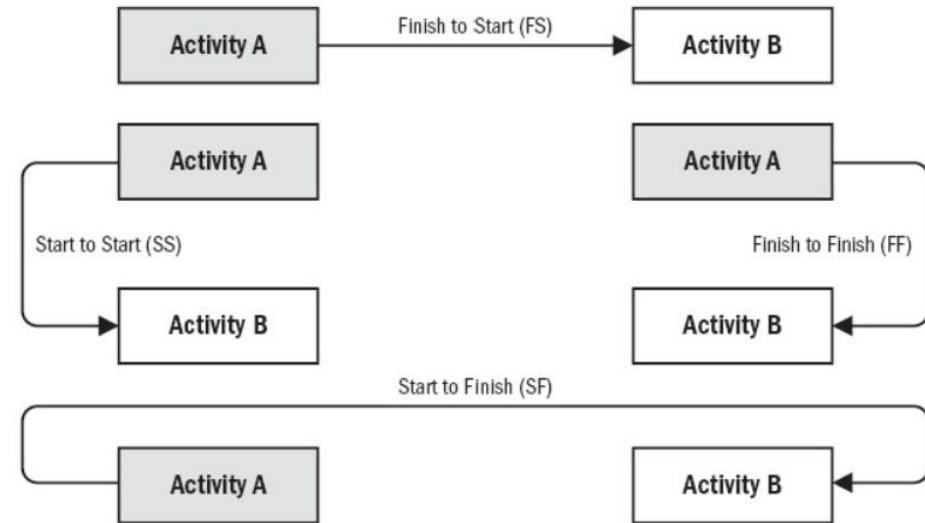
$$t_{\text{beta}} = (a + 4m + b)/6$$



- Very useful to also have a measure of **uncertainty**

Sequence Activities

- Relationships
- Dependencies
- Resource constraints
- Milestones

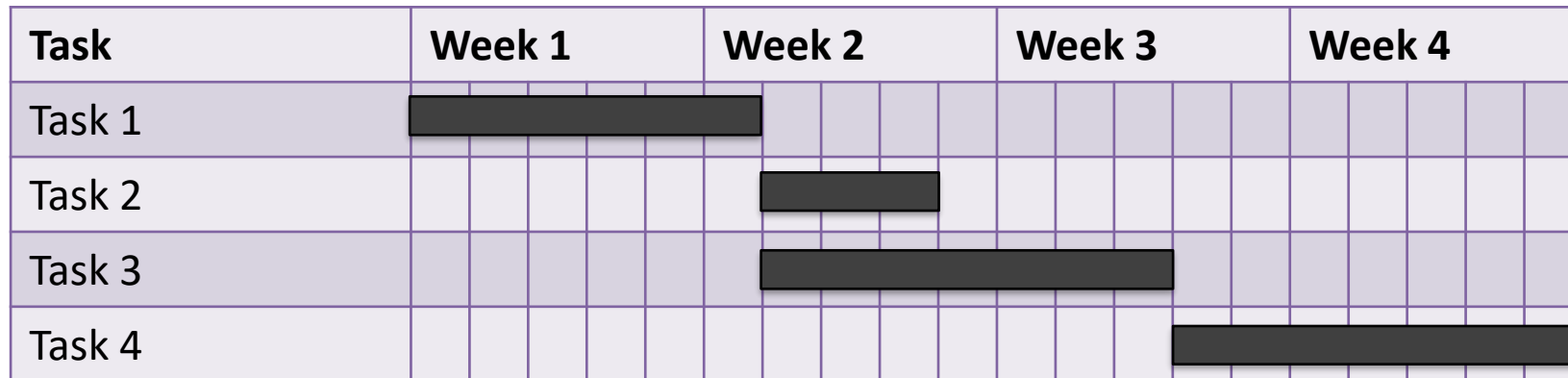


PMBOK Guide: Precedence Diagramming Method

Gantt Charts

Graphical Visualisation of the project, showing:

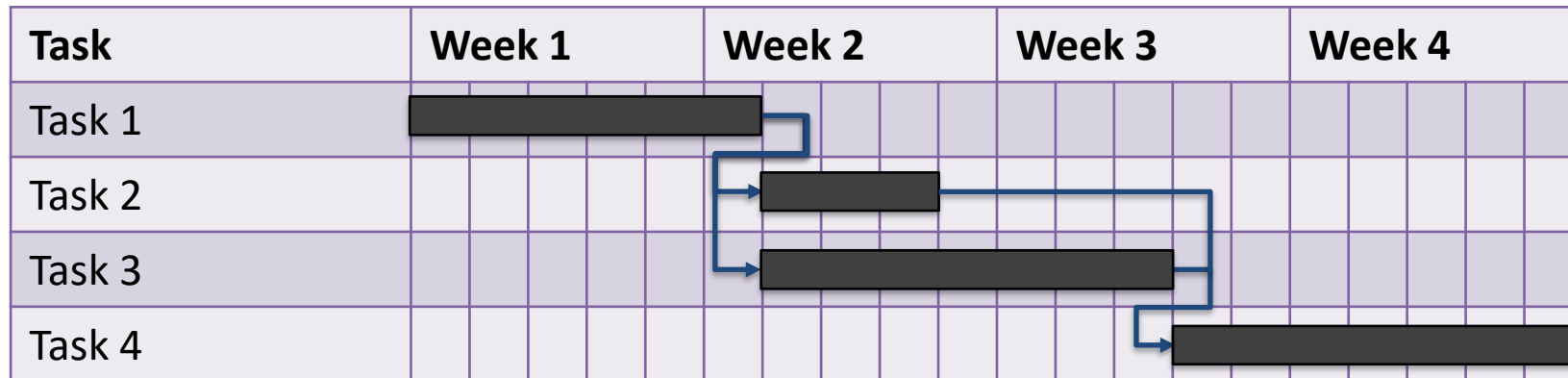
- **temporal** schedule of activities



Gantt Charts

Graphical Visualisation of the project, showing:

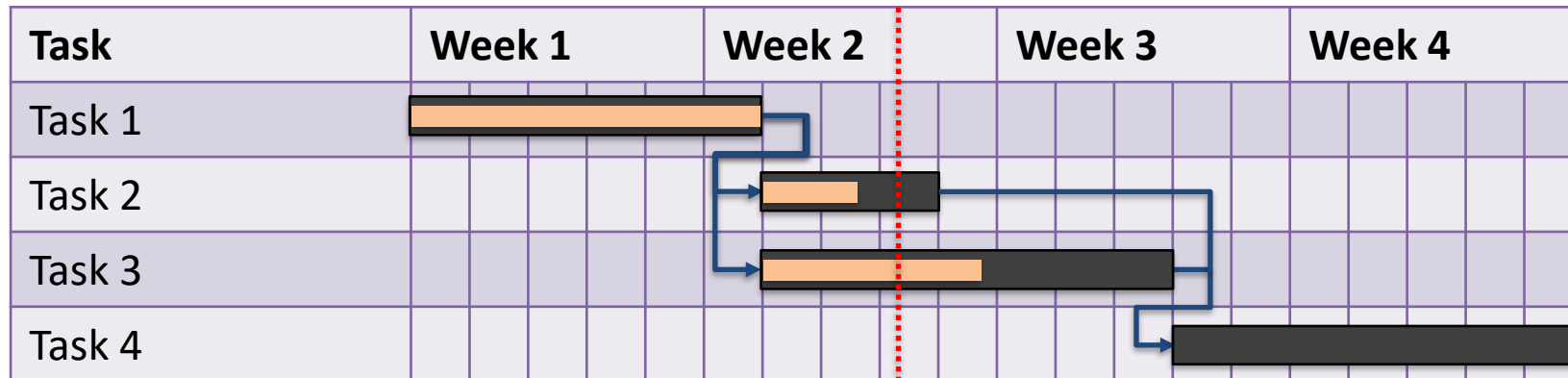
- **temporal** schedule of activities
- **dependencies** between activities



Gantt Charts

Graphical Visualisation of the project, showing:

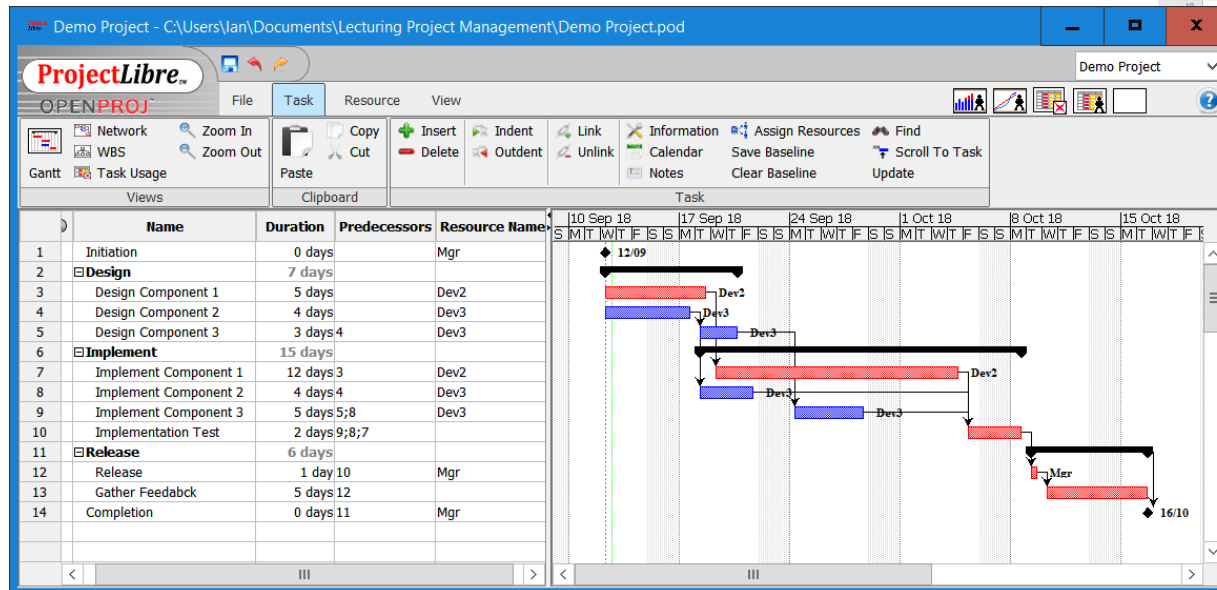
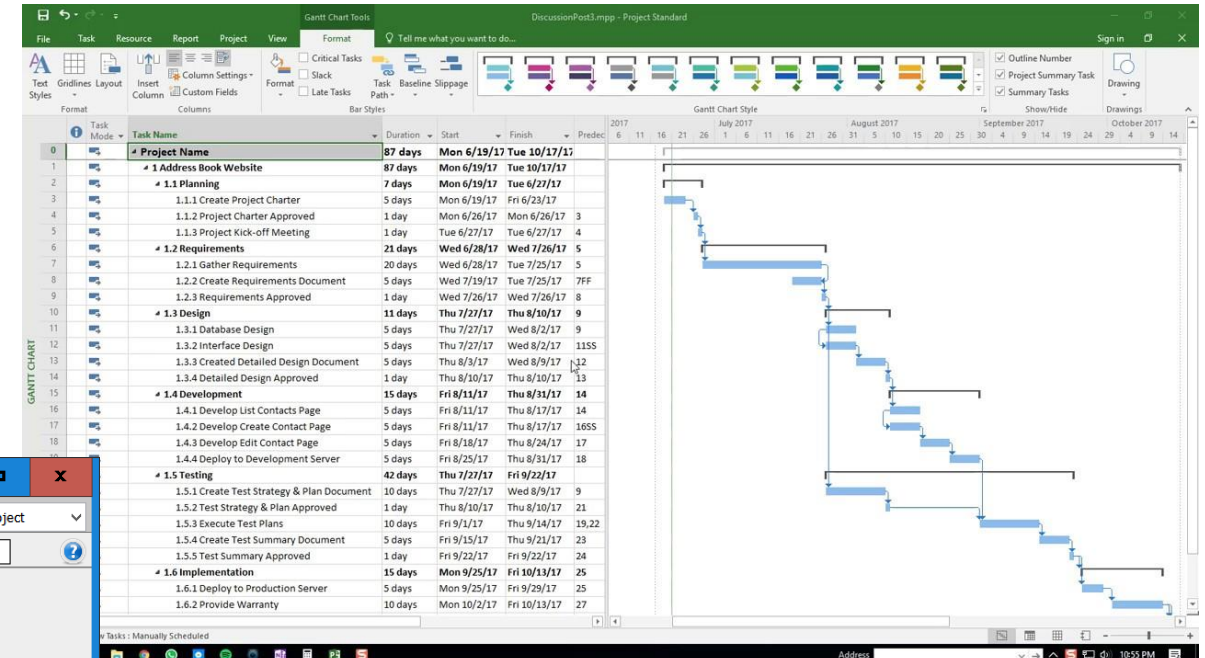
- **temporal** schedule of activities
- **dependencies** between activities
- **progress** of activities



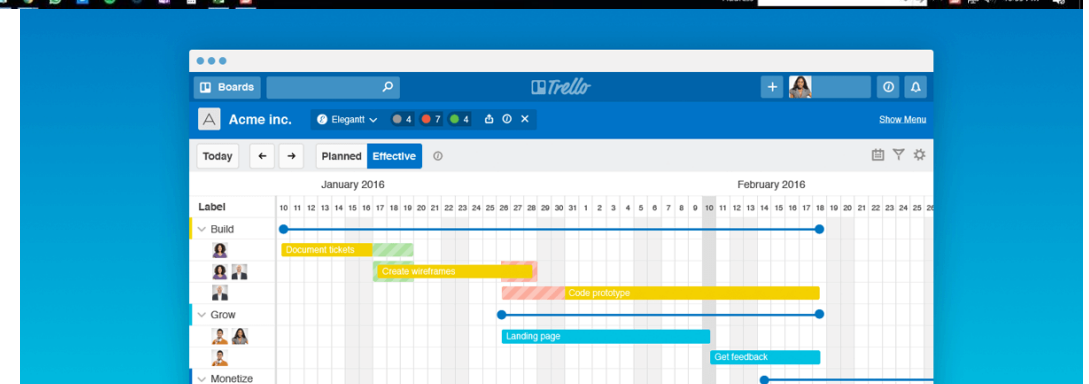
Gantt Tools

- Online tools:
 - Asana with the InstaGantt plug-in
 - Trello with the EleGantt plug-in
- Others:
 - Project Libre
 - Microsoft Project
 - ...and many others

Microsoft Project



Project Libre



Elegantt

Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - **Project Network Diagram**
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)

Project Network Diagram (PND)

- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Project Network Diagram (PND)

- *Project Network Diagram (PND)*: A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Project Network Diagram (PND)

- *Project Network Diagram (PND)*: A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

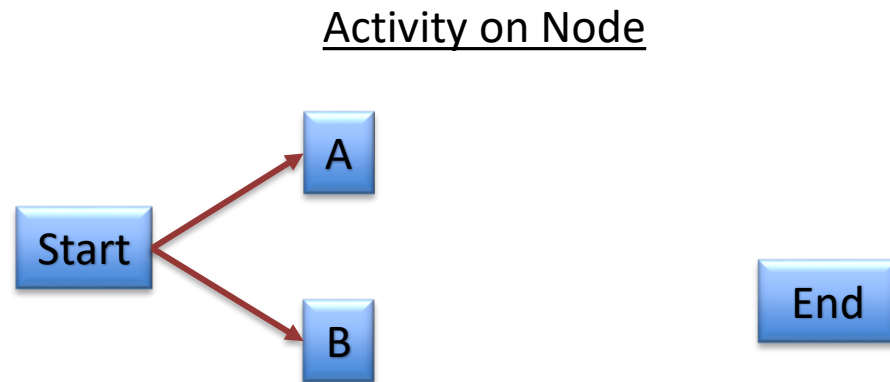
Activity on Node



Project Network Diagram (PND)

- *Project Network Diagram (PND)*: A graphical way to view a project's tasks, dependencies and the **critical path**.

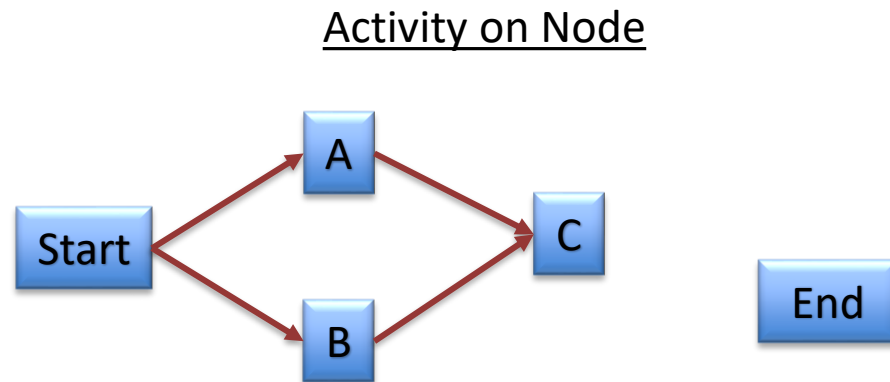
Task	Dependencies
A	-
B	-
C	A, B
D	B



Project Network Diagram (PND)

- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

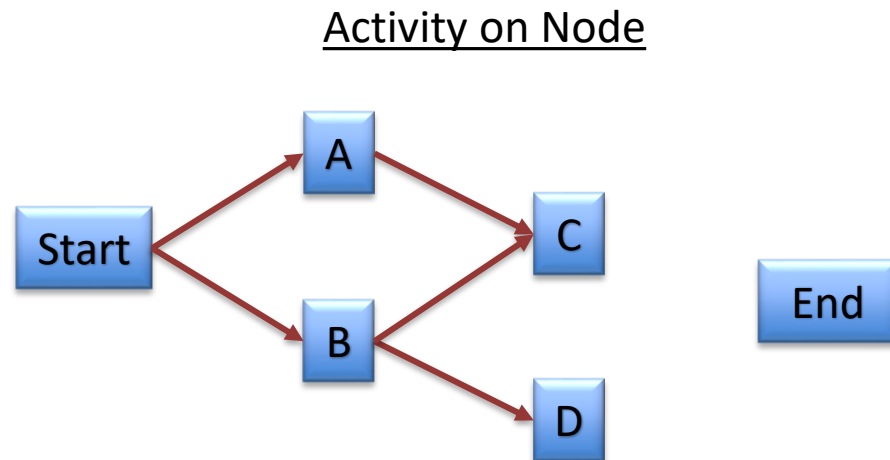
Task	Dependencies
A	-
B	-
C	A, B
D	B



Project Network Diagram (PND)

- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

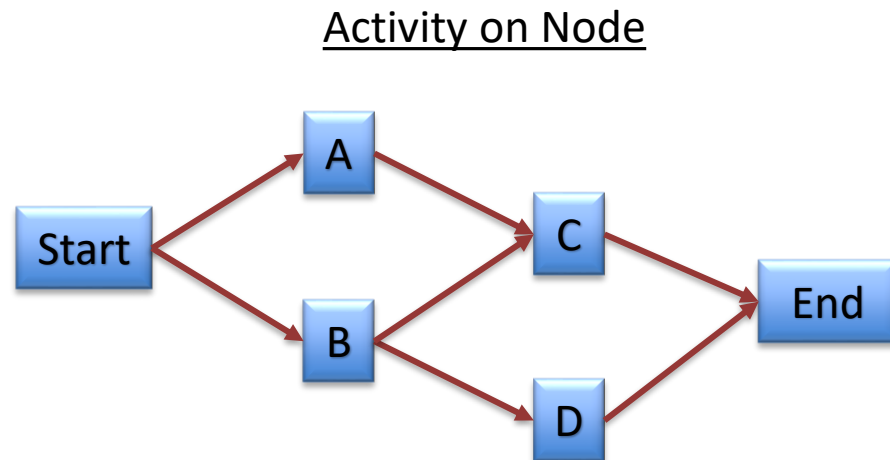
Task	Dependencies
A	-
B	-
C	A, B
D	B



Project Network Diagram (PND)

- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

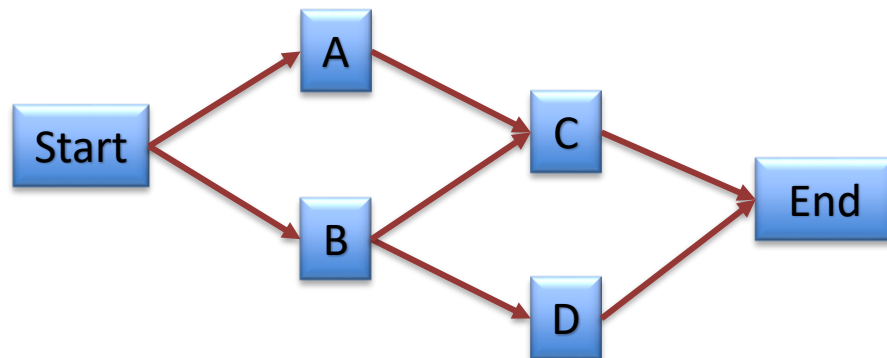


Project Network Diagram (PND)

- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

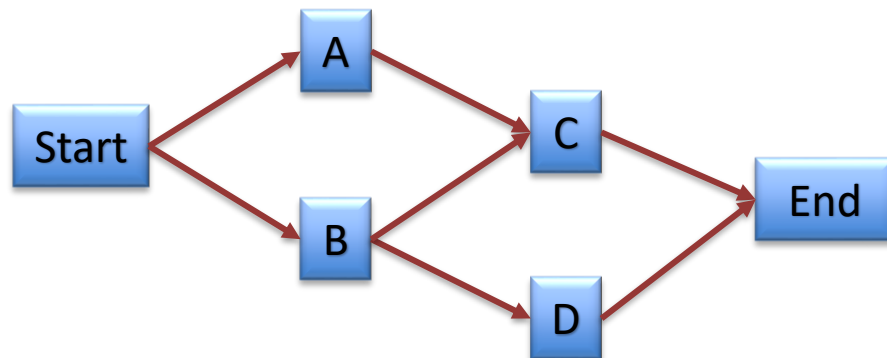


Project Network Diagram (PND)

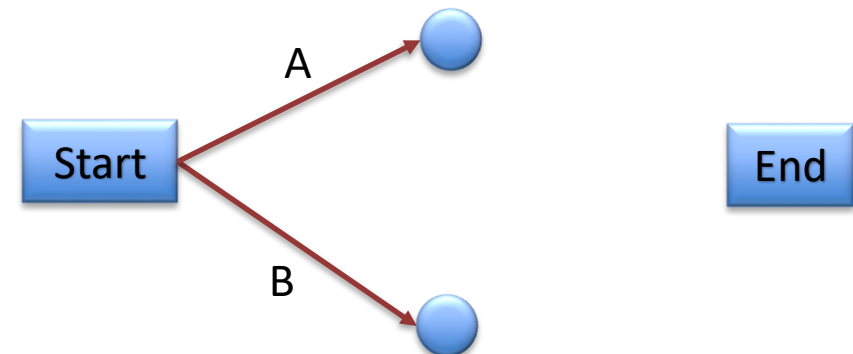
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

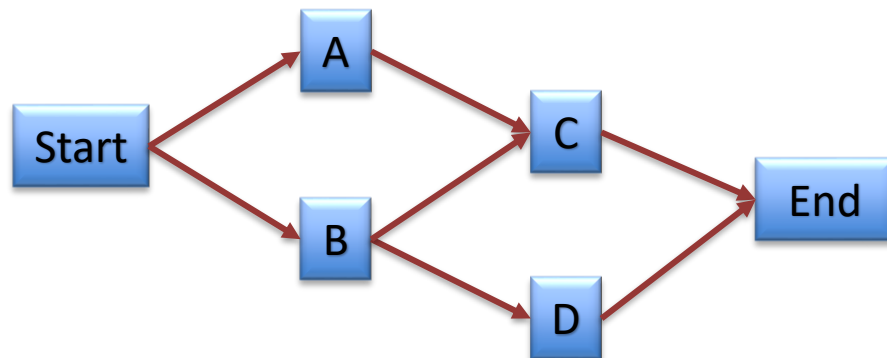


Project Network Diagram (PND)

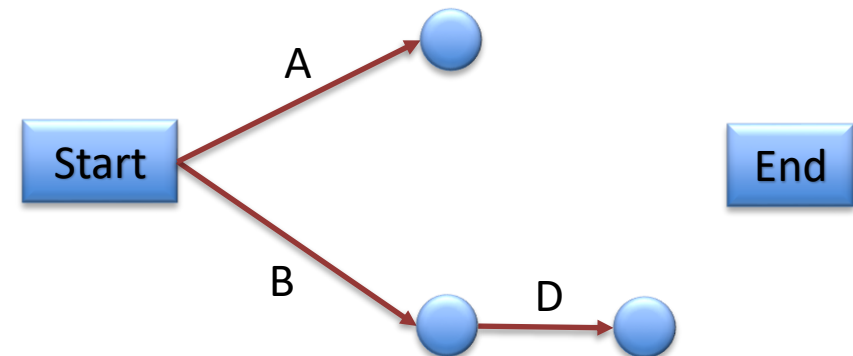
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

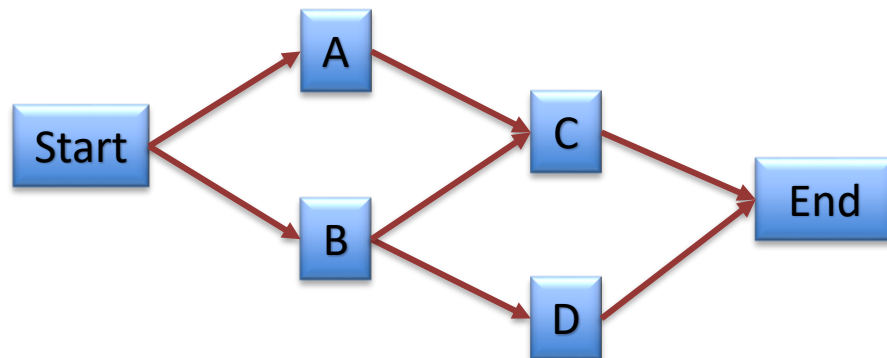


Project Network Diagram (PND)

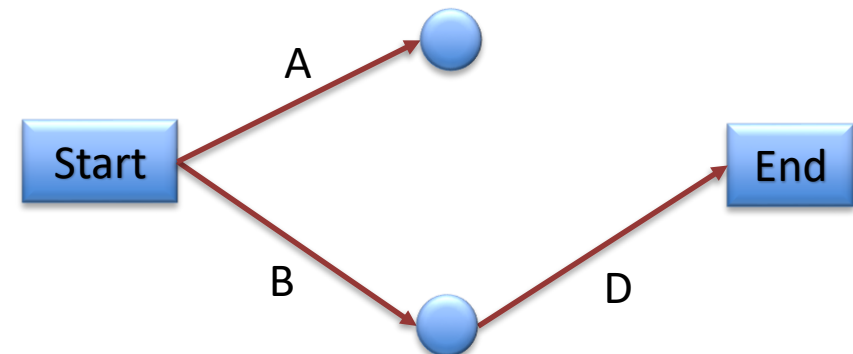
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

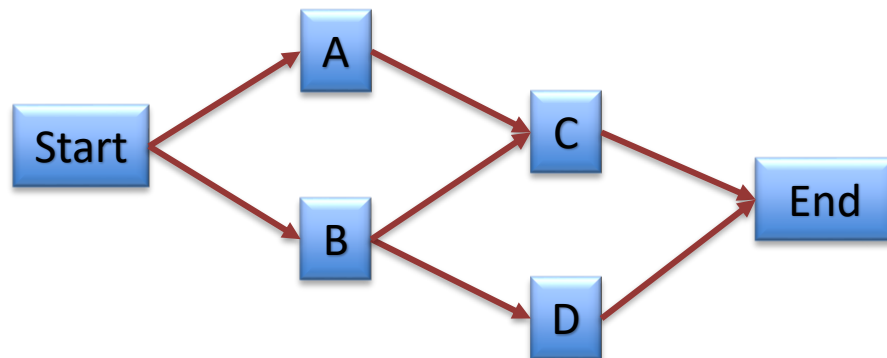


Project Network Diagram (PND)

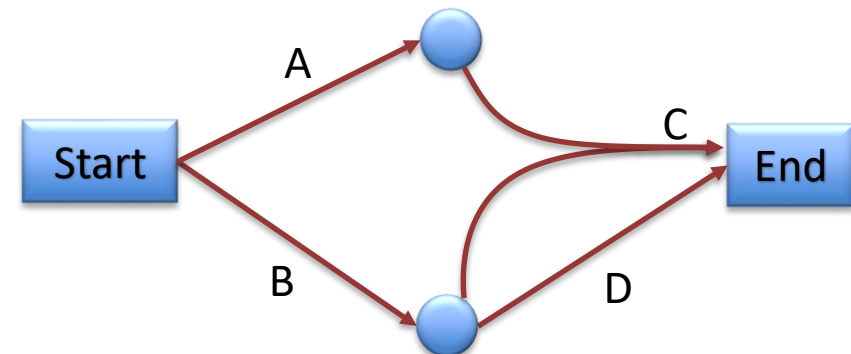
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

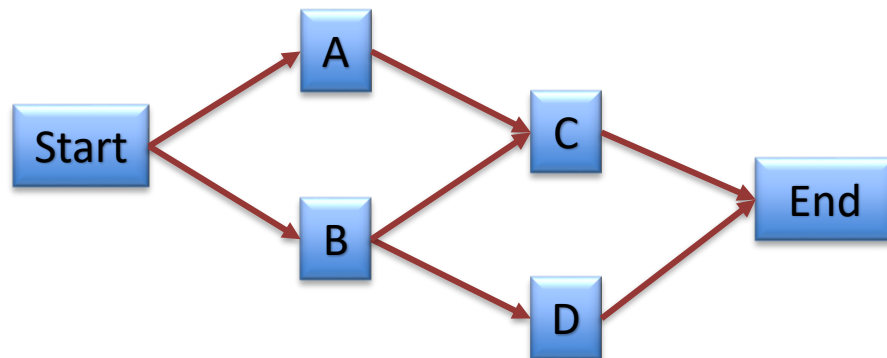


Project Network Diagram (PND)

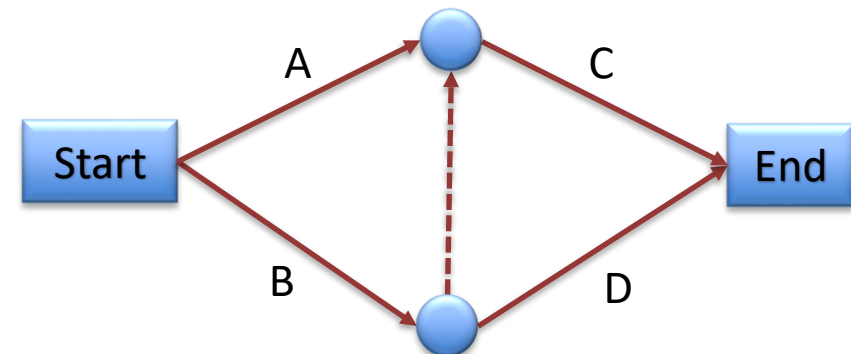
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow

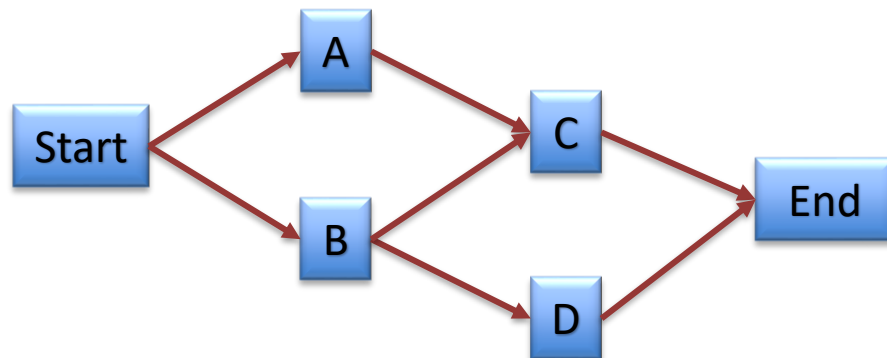


Project Network Diagram (PND)

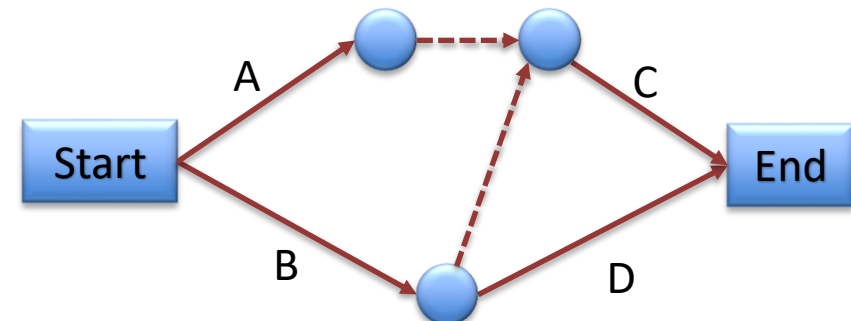
- *Project Network Diagram (PND):* A graphical way to view a project's tasks, dependencies and the **critical path**.

Task	Dependencies
A	-
B	-
C	A, B
D	B

Activity on Node



Event on Node/Activity on Arrow



Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - **Critical Path Method (CPM)**
 - Program Evaluation and Review Technique (PERT)

Critical Path

Critical path(s): A sequence of activities starting from the first activity of the project and ending with the last. Activities on a critical path cannot be delayed without extending the project duration.

Critical Path Method (CPM)

Duration (D):	The duration of the activity
Earliest Start (ES):	Earliest time an activity can start (i.e. once previous activities over) ES = Maximum EF from immediate predecessors
Earliest Finish (EF):	EF = ES + D
Latest Finish (LF):	Latest time an activity can finish without delaying the project LF = Minimum LS from immediate successors
Latest Start (LS):	LS = LF - D
Total Float (TF):	Amount of time a task can be delayed without delaying the project TF = LS – ES = LF – EF

Critical Path Method (CPM)

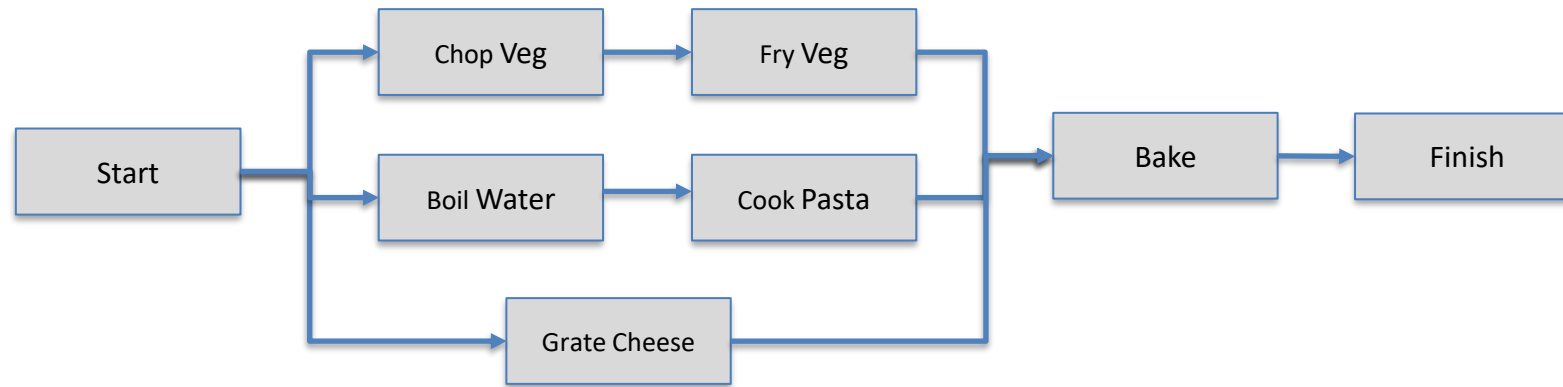
Algorithm:

1. Construct PND
2. Forward Pass:
 - $ES = \text{Maximum } EF \text{ from immediate predecessors}$
 - $EF = ES + D$
3. Backward Pass:
 - $LF = \text{Minimum } LS \text{ from immediate successors}$
 - $LS = LF - D$
 - $TF = LS - ES = LF - EF$

Critical Path Method (CPM)

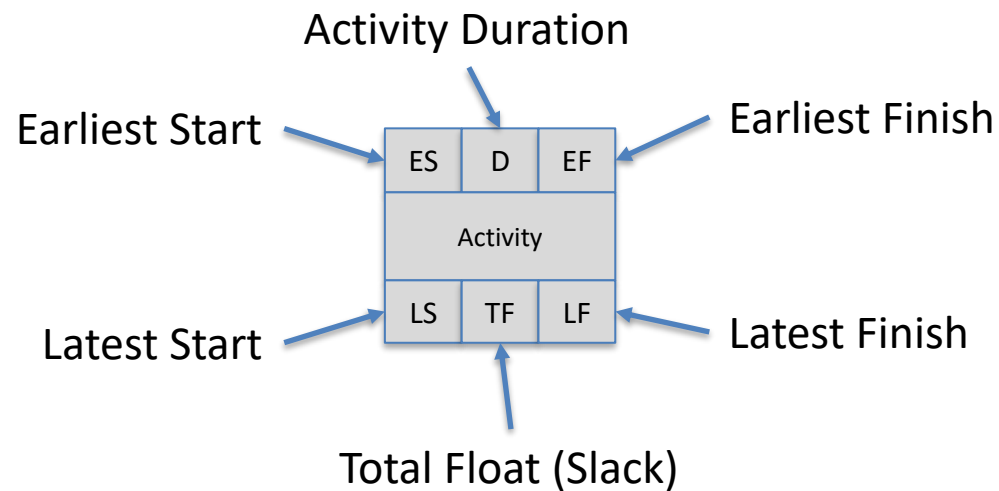
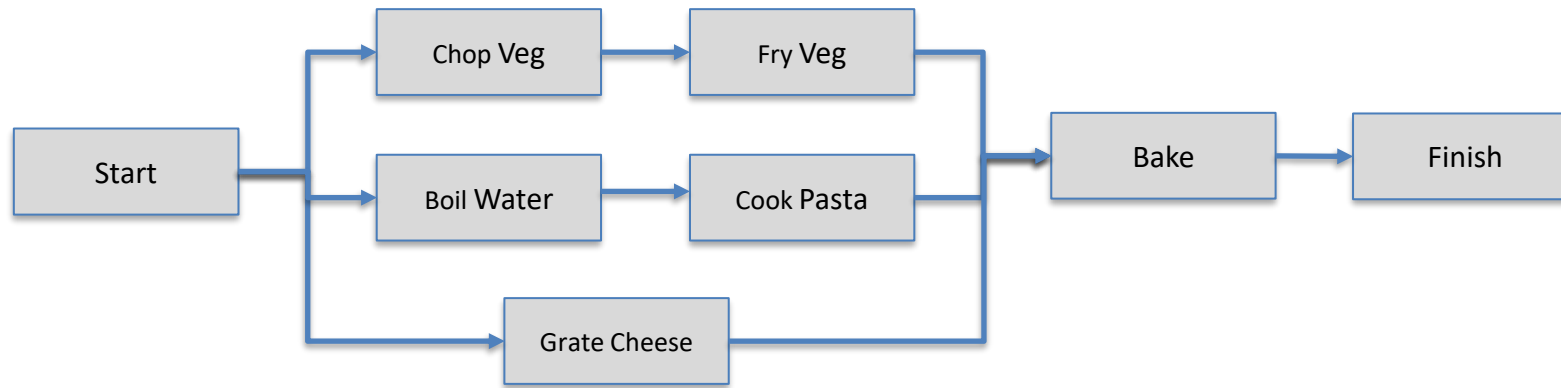
ID	Name	Duration	Dependencies
A	Chop Veg	5	-
B	Fry Veg	10	A
C	Boil Water	4	-
D	Cook Pasta	20	C
E	Grate Cheese	3	-
F	Bake	30	B,D,E

Critical Path Method (CPM)



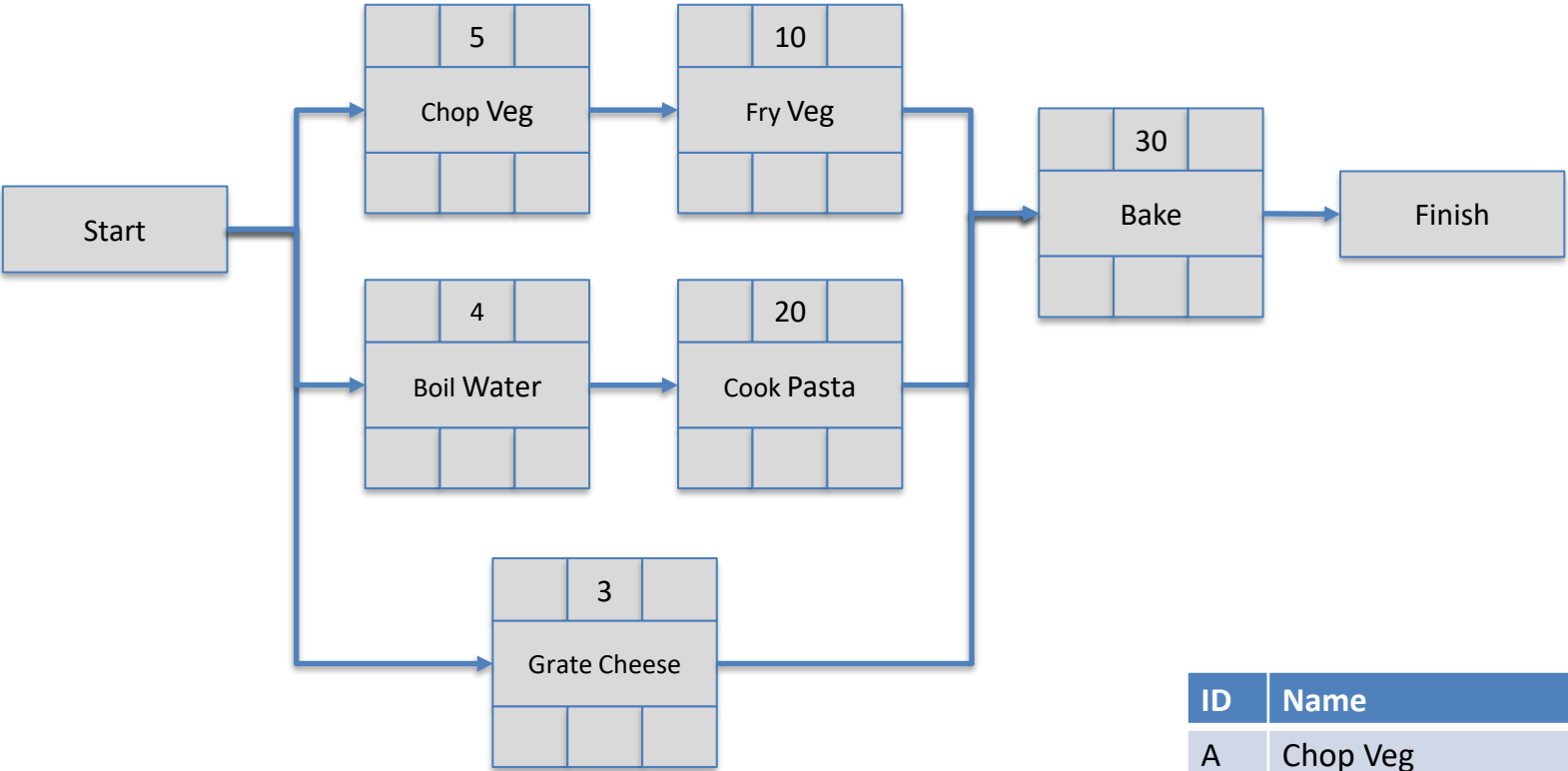
ID	Name	Duration	Dependencies
A	Chop Veg	5	-
B	Fry Veg	10	A
C	Boil Water	4	-
D	Cook Pasta	20	C
E	Grate Cheese	3	-
F	Bake	30	B,D,E

Critical Path Method (CPM)



ID	Name	Duration	Dependencies
A	Chop Veg	5	-
B	Fry Veg	10	A
C	Boil Water	4	-
D	Cook Pasta	20	C
E	Grate Cheese	3	-
F	Bake	30	B,D,E

Critical Path Method (CPM)

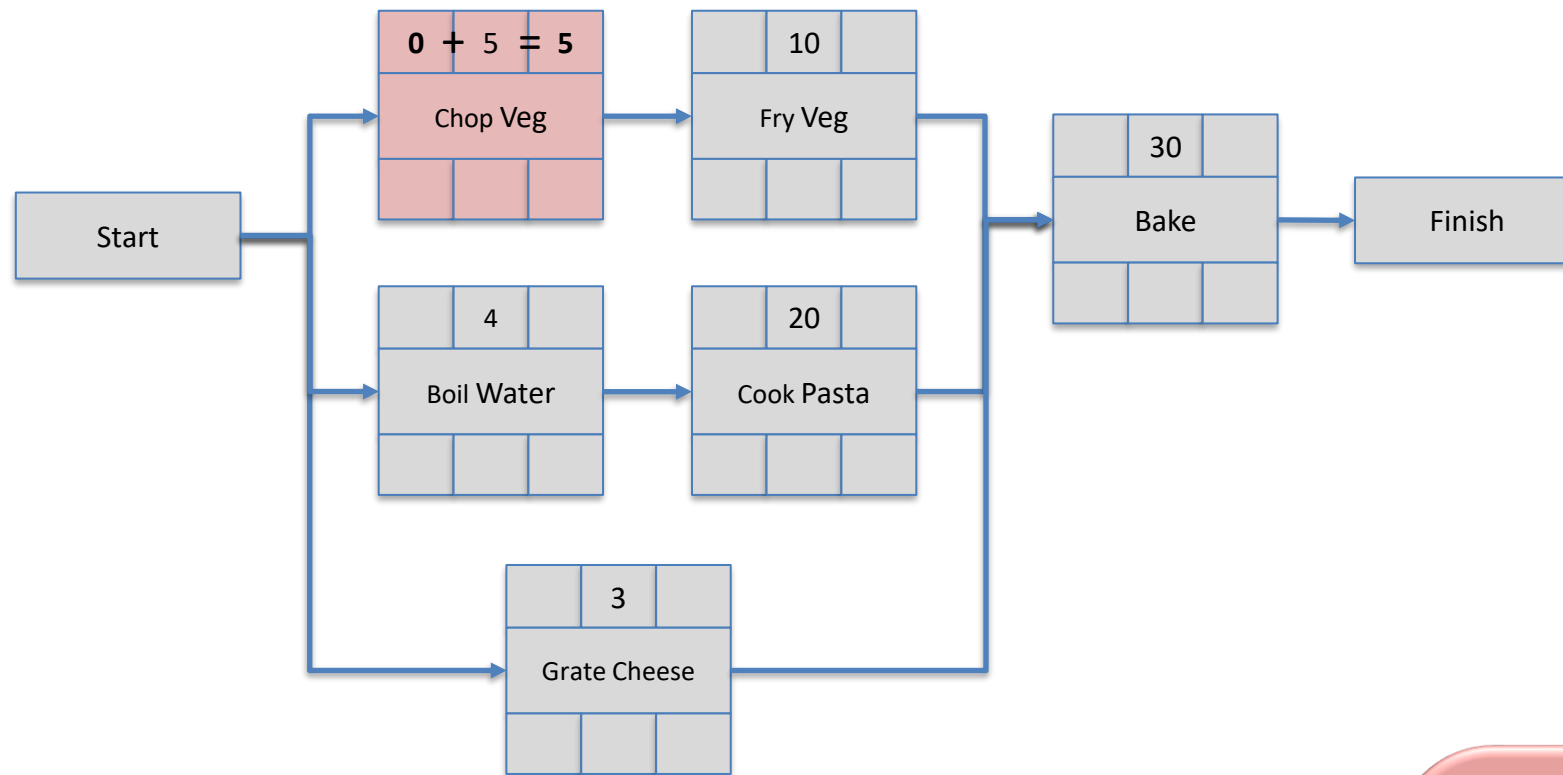


Key:

ES	D	EF
Activity		
LS	TF	LF

ID	Name	Duration	Dependencies
A	Chop Veg	5	-
B	Fry Veg	10	A
C	Boil Water	4	-
D	Cook Pasta	20	C
E	Grate Cheese	3	-
F	Bake	30	B,D,E

Critical Path Method (CPM)



Key:

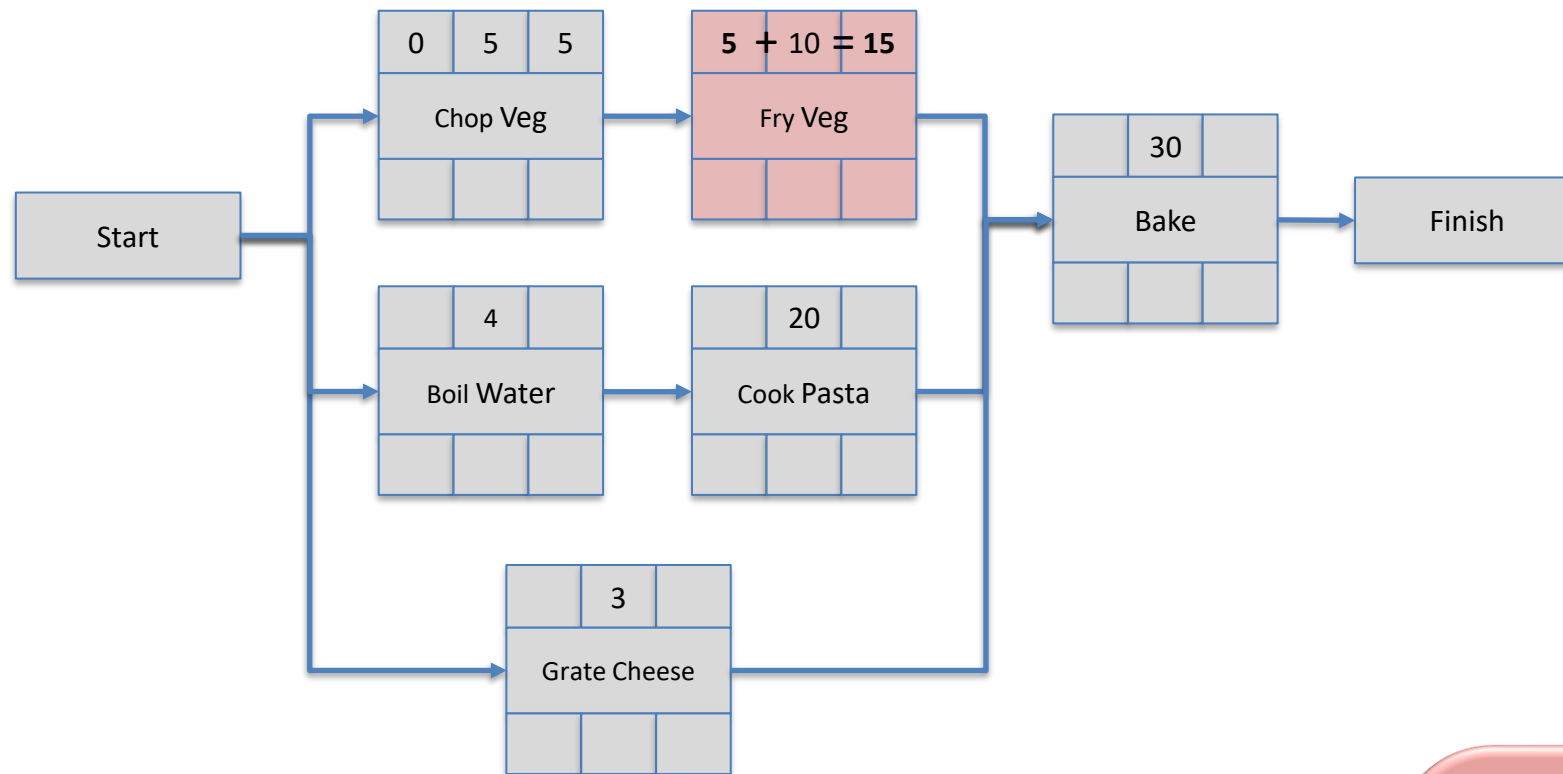
ES	D	EF
Activity		
LS	TF	LF

Forward Pass

ES = Max EF from predecessors

EF = ES + D

Critical Path Method (CPM)



Key:

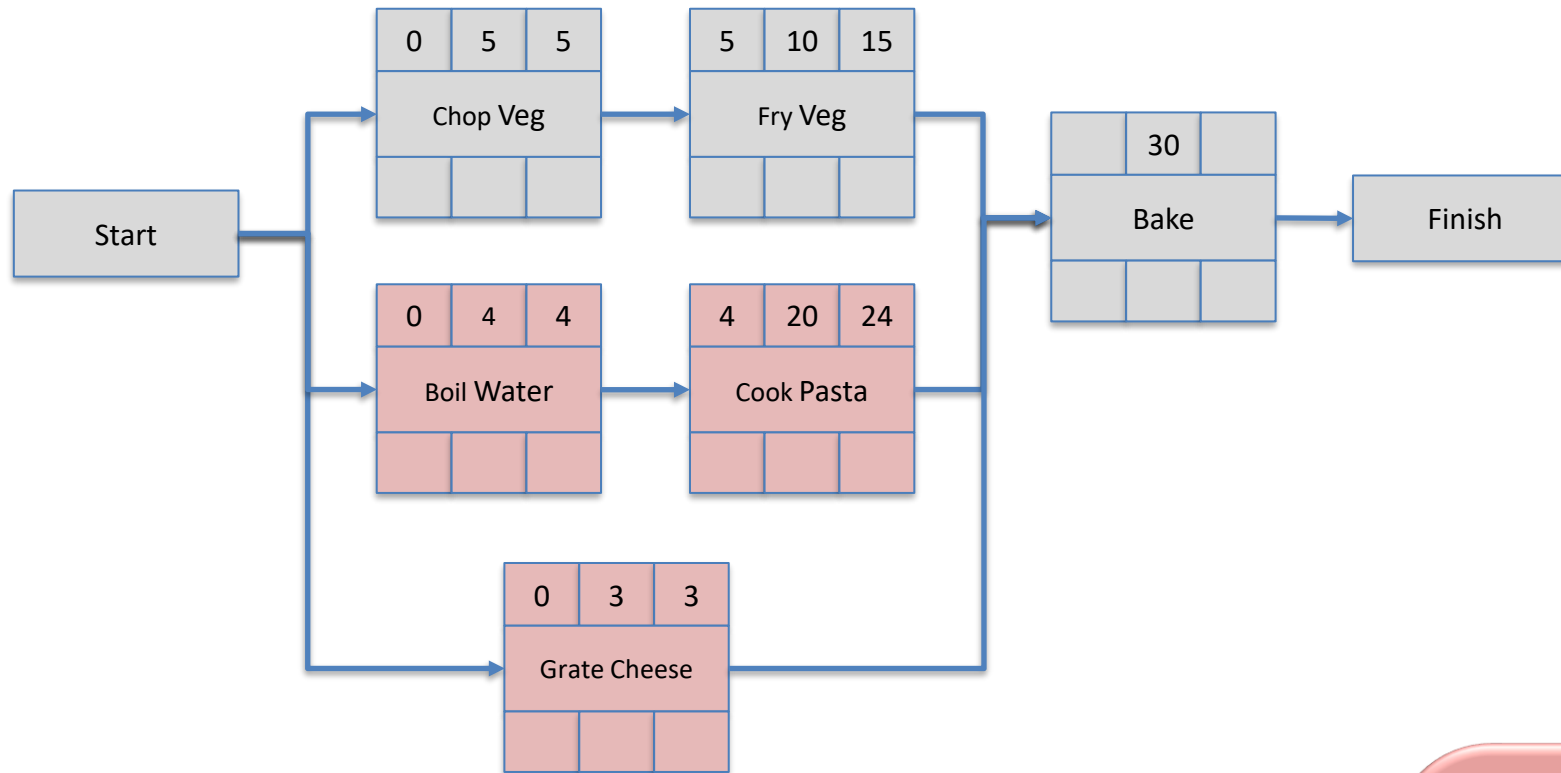
ES	D	EF
Activity		
LS	TF	LF

Forward Pass

ES = Max EF from predecessors

EF = ES + D

Critical Path Method (CPM)



Key:

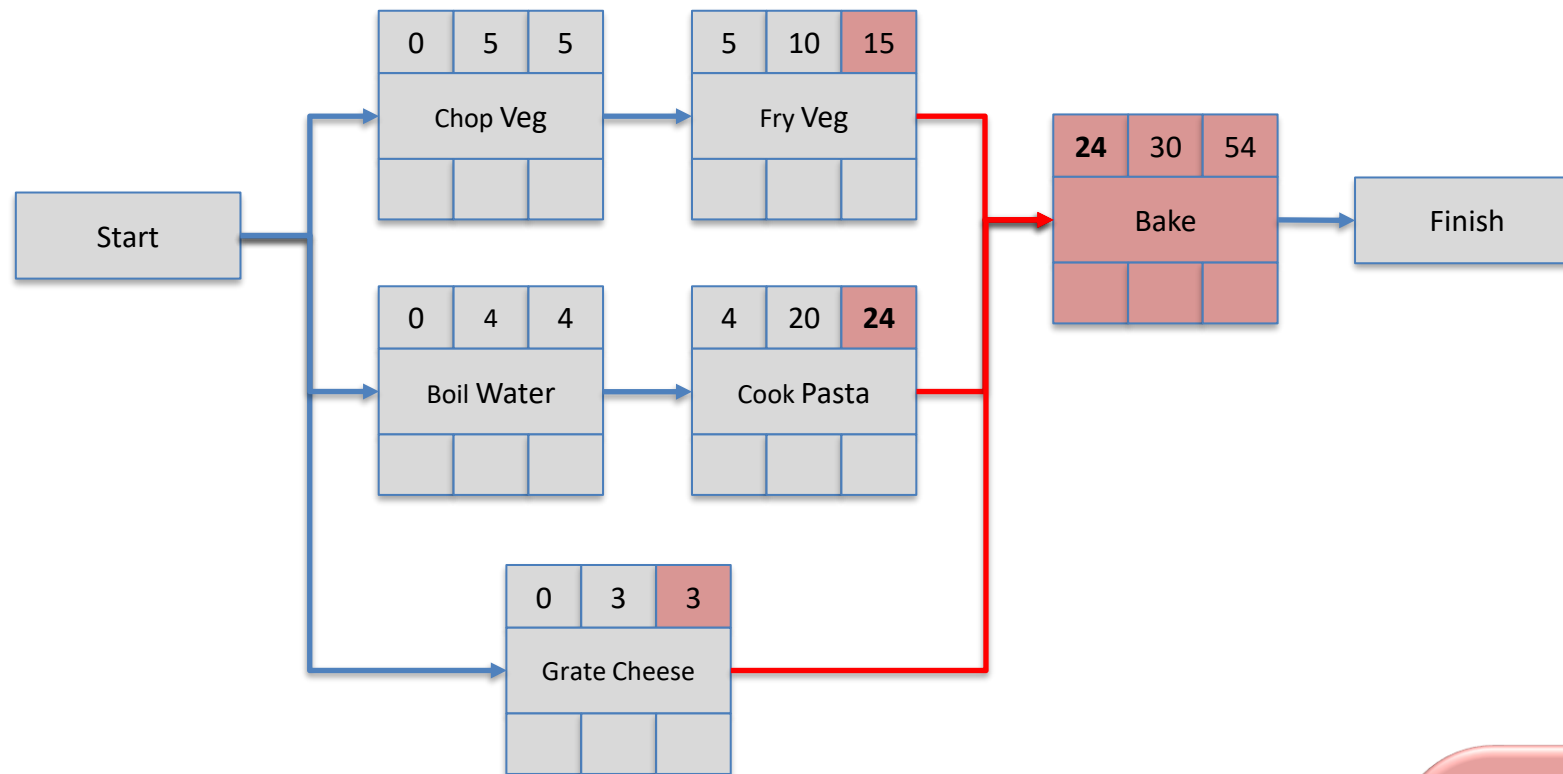
ES	D	EF
Activity		
LS	TF	LF

Forward Pass

ES = Max EF from predecessors

EF = ES + D

Critical Path Method (CPM)



Key:

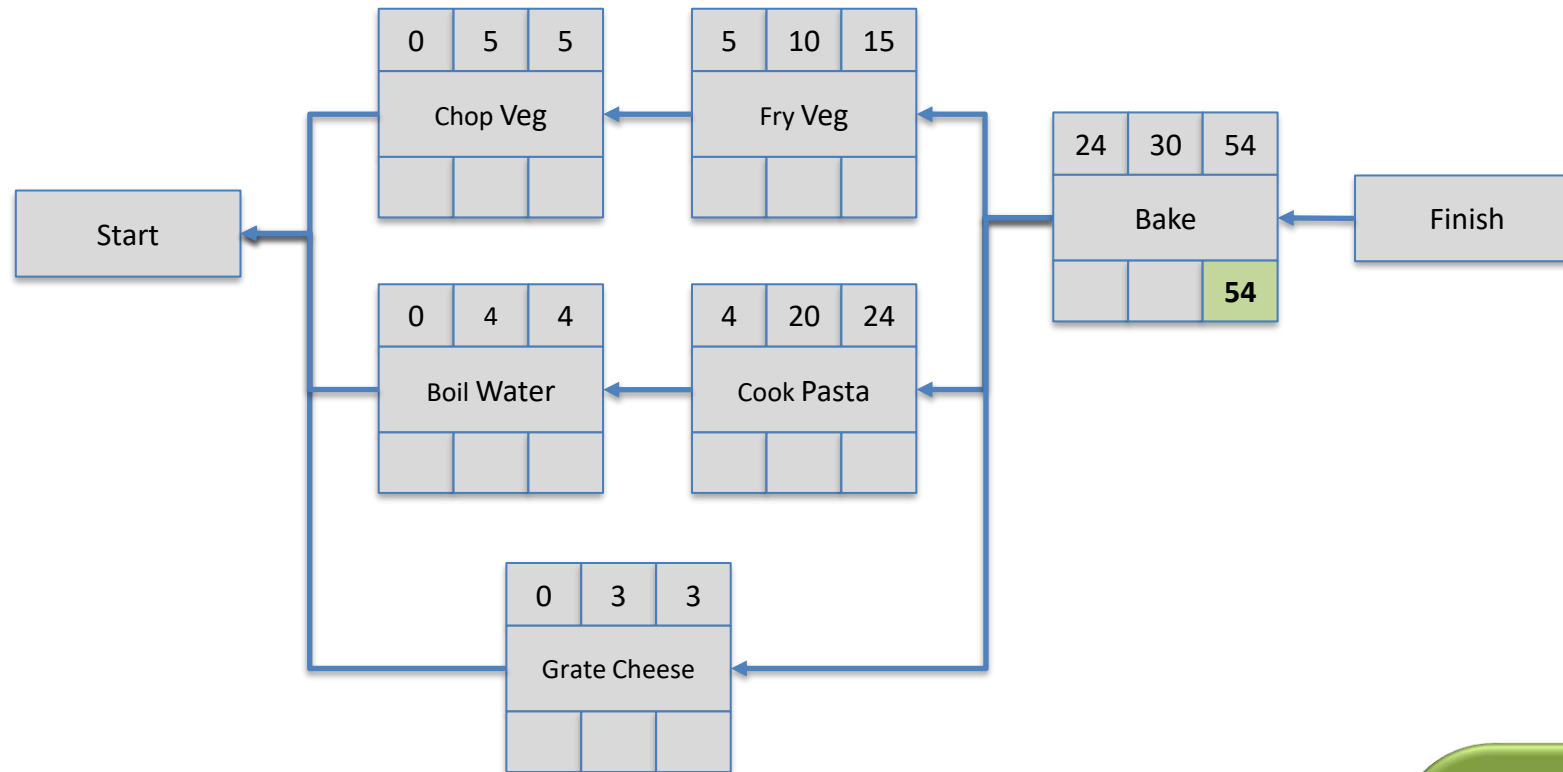
ES	D	EF
Activity		
LS	TF	LF

Forward Pass

ES = Max EF from predecessors

EF = ES + D

Critical Path Method (CPM)



Key:

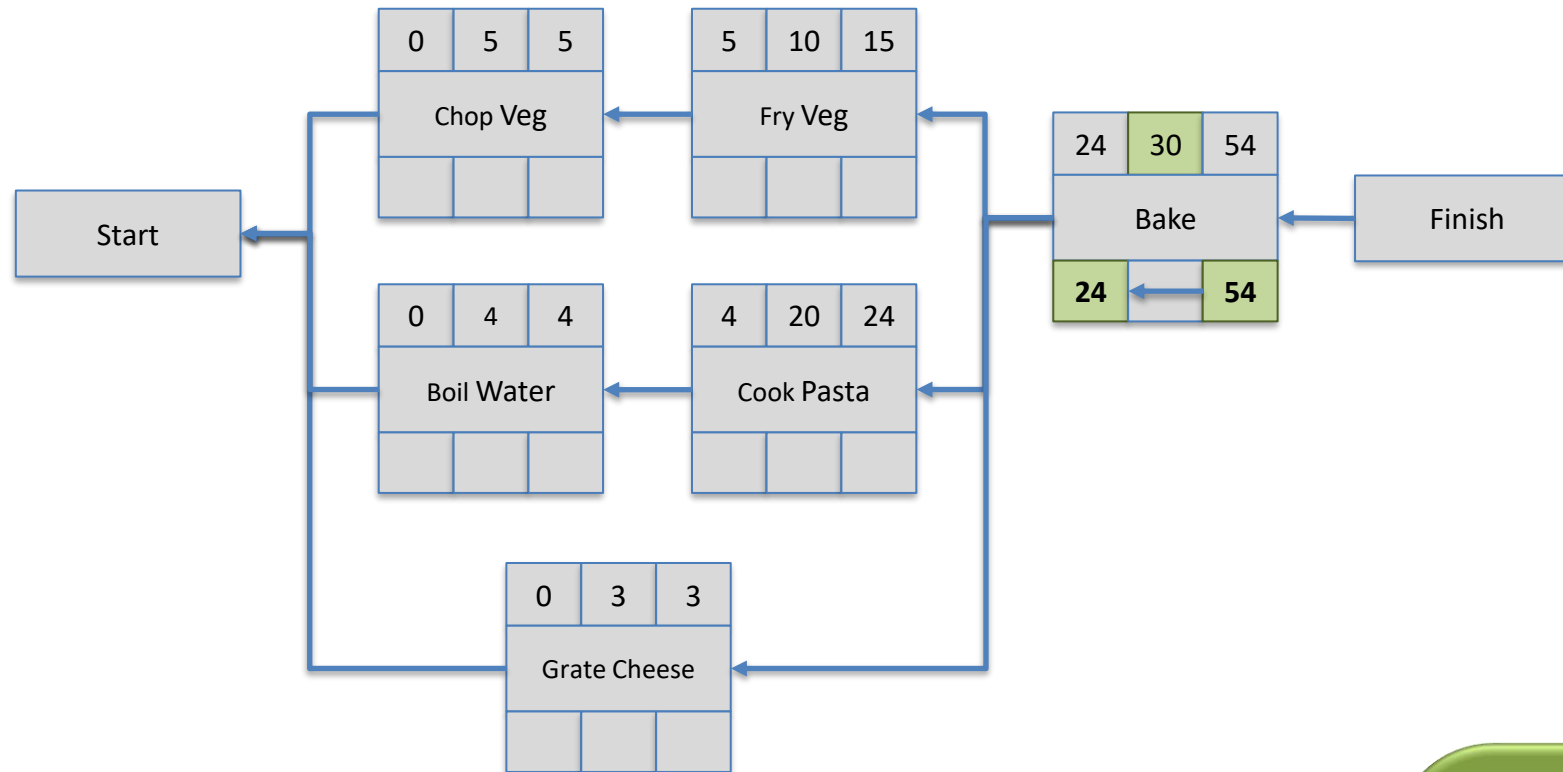
ES	D	EF
Activity		
LS	TF	LF

Backward Pass

LF = Min LS from successors

LS = LF – D

Critical Path Method (CPM)



Key:

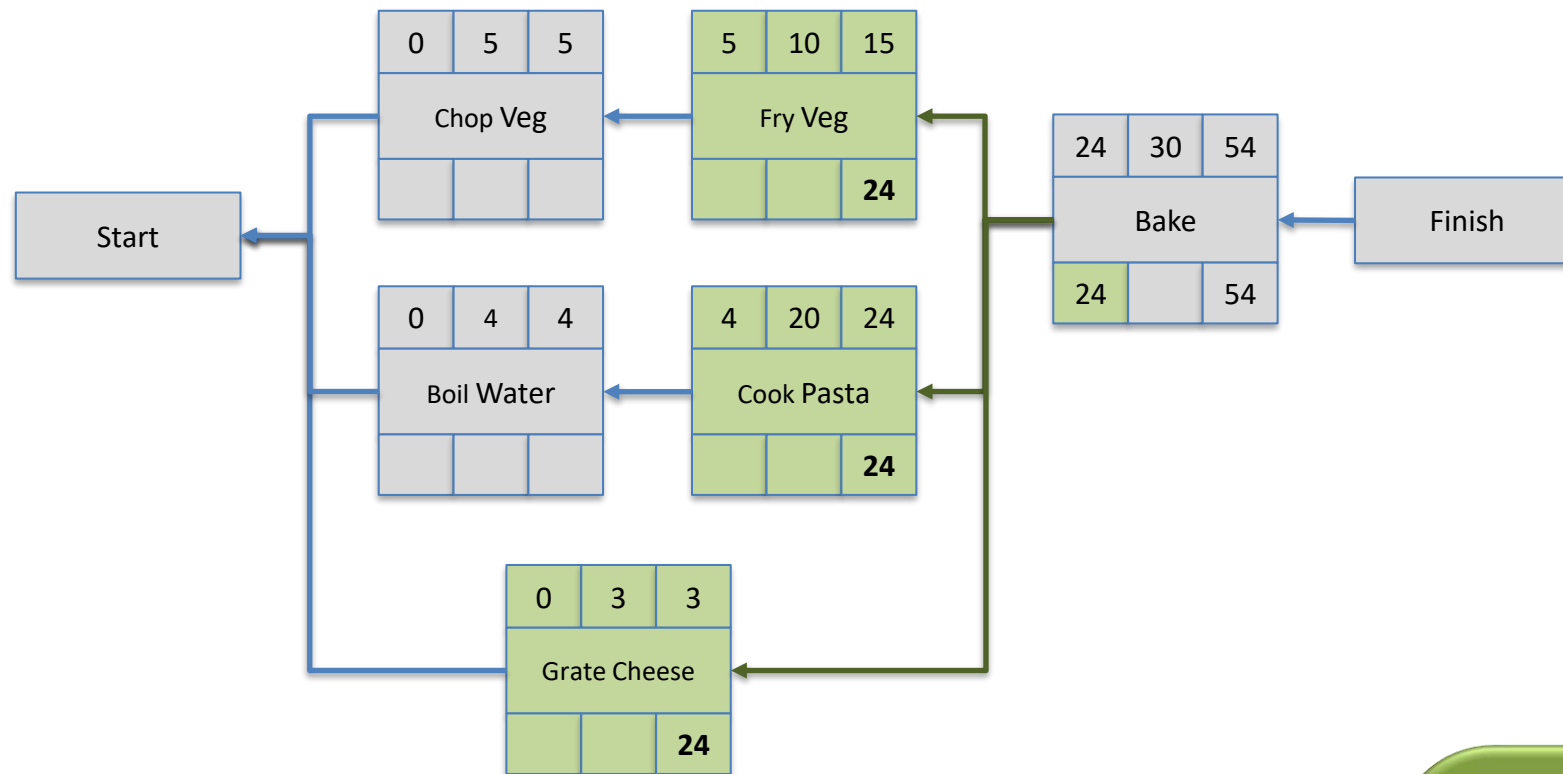
ES	D	EF
Activity		
LS	TF	LF

Backward Pass

LF = Min LS from successors

LS = LF - D

Critical Path Method (CPM)



Key:

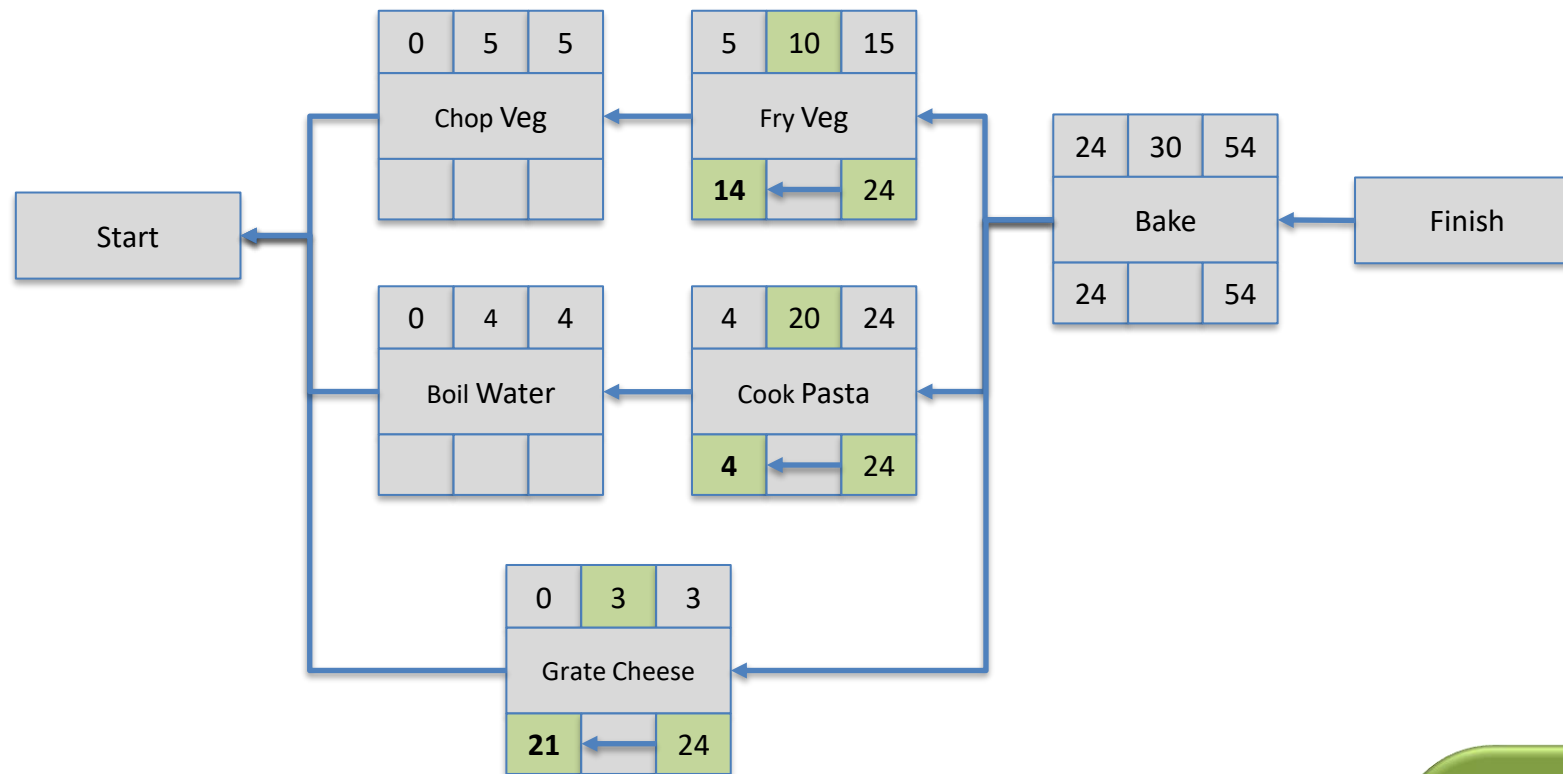
ES	D	EF
Activity		
LS	TF	LF

Backward Pass

LF = Min LS from successors

LS = LF - D

Critical Path Method (CPM)



Key:

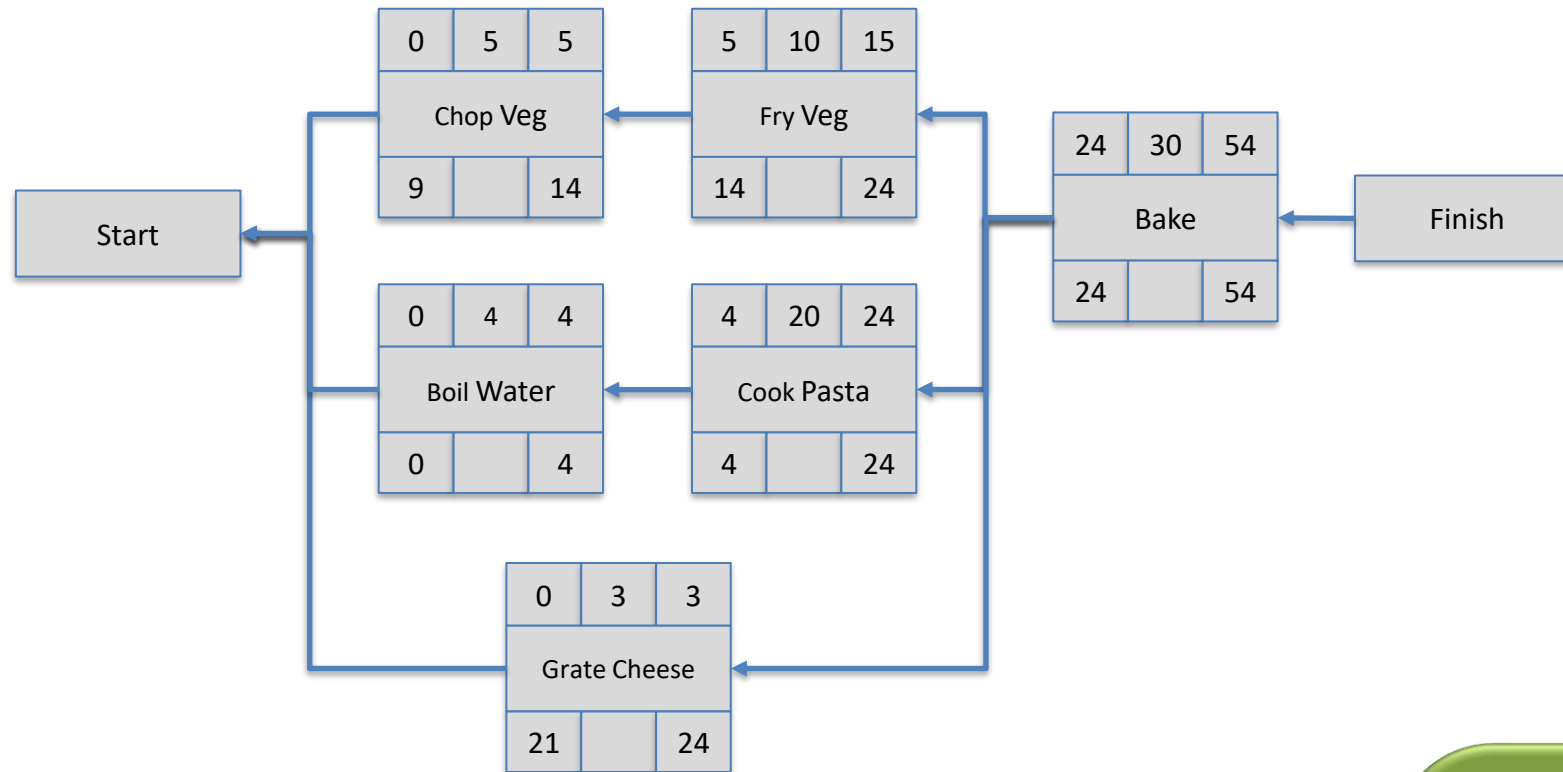
ES	D	EF
Activity		
LS	TF	LF

Backward Pass

LF = Min LS from successors

LS = LF - D

Critical Path Method (CPM)



Key:

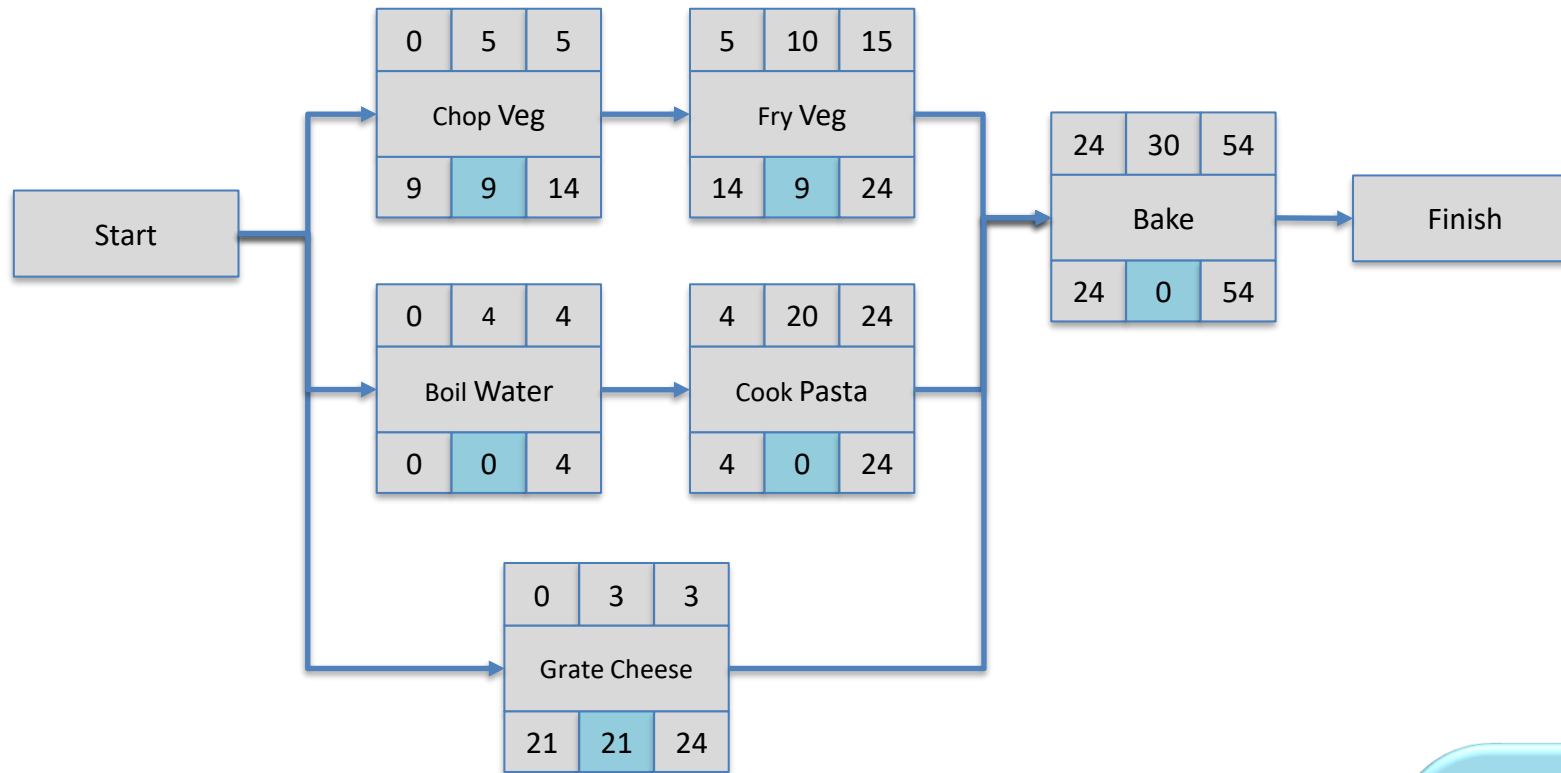
ES	D	EF
Activity		
LS	TF	LF

Backward Pass

LF = Min LS from successors

LS = LF – D

Critical Path Method (CPM)



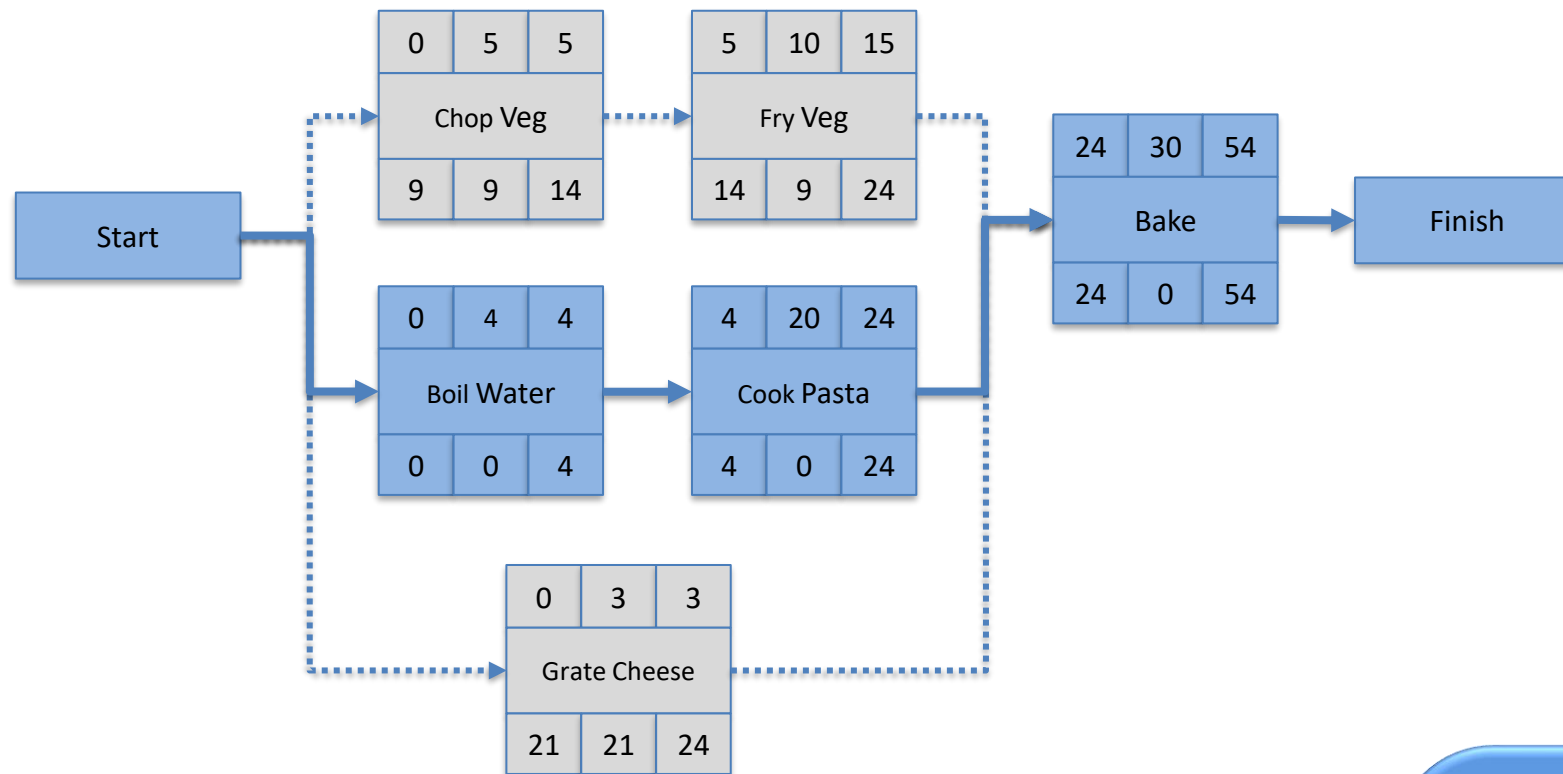
Key:

ES	D	EF
Activity		
LS	TF	LF

Total Float (Slack)

$$F = LS - ES = LF - EF$$

Critical Path Method (CPM)



Key:

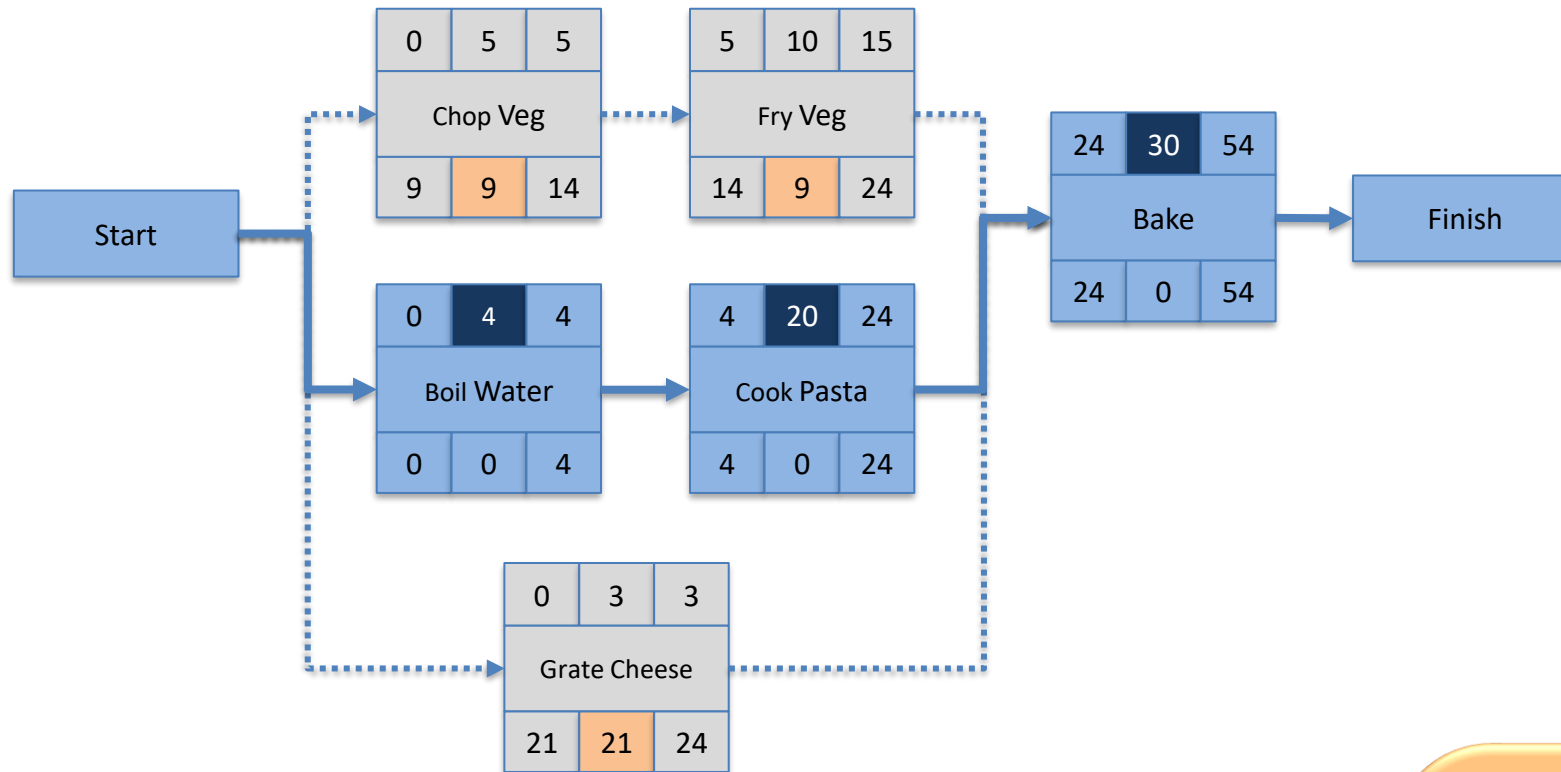
ES	D	EF
Activity		
LS	TF	LF

Critical Path
Path without any slack

CPM: Drag Time

- Critical Path:** The sequence(s) of activities from start to finish that can't be delayed without delaying the project.
- Critical Task:** Can't be delayed without delaying the project (i.e. no slack).
- Drag time:** The amount of time a critical task adds to the project duration (the time it would need to be shortened by to no longer be critical)
The minimum of: **D** or the minimum **TF** of parallel tasks

CPM: Drag Time



Key:

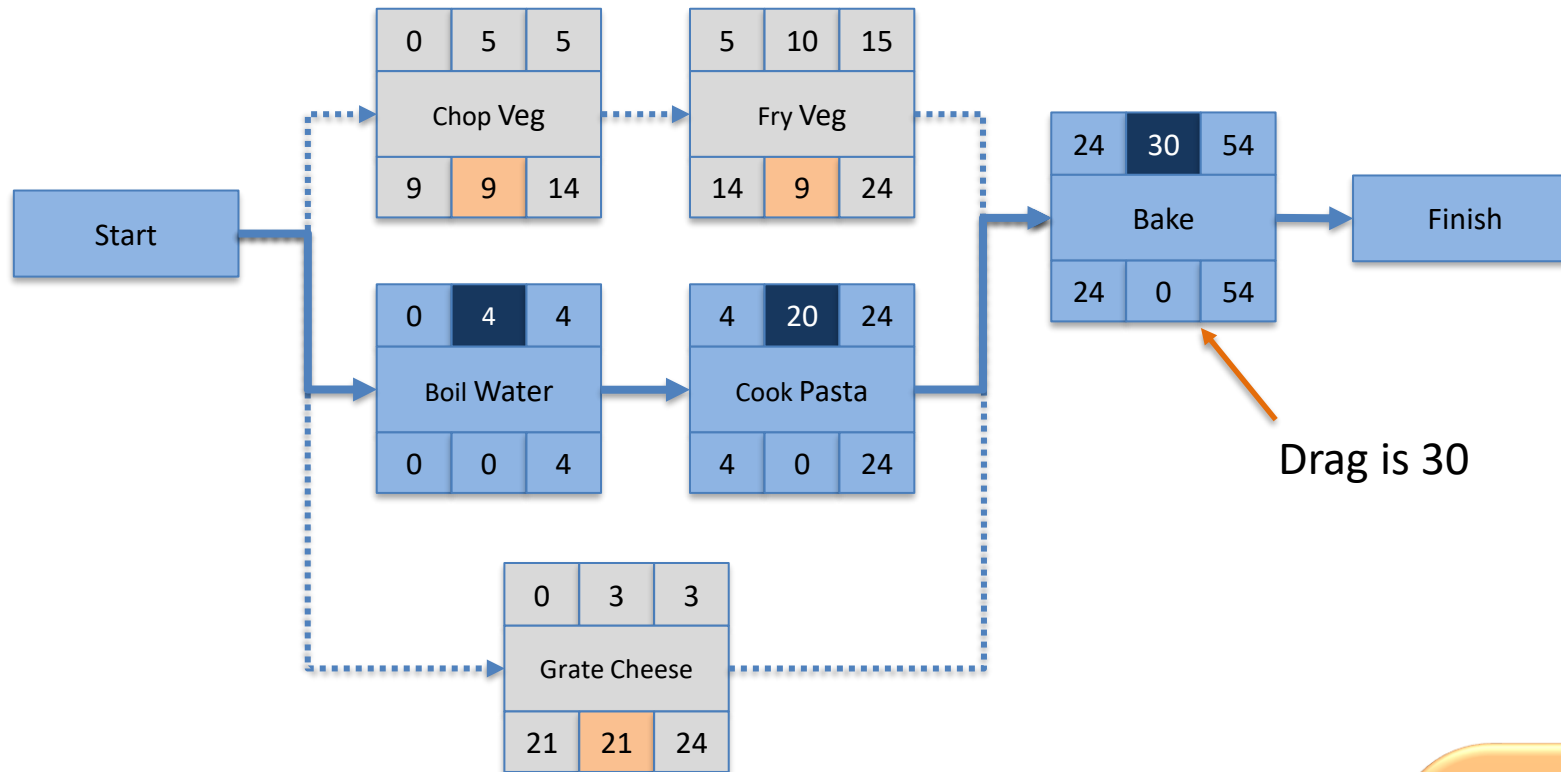
ES	D	EF
Activity		
LS	TF	LF

Drag

Amount of time a critical task delays the project.

$$= \min(D , \min(TF_{\text{parallel}}))$$

CPM: Drag Time



Key:

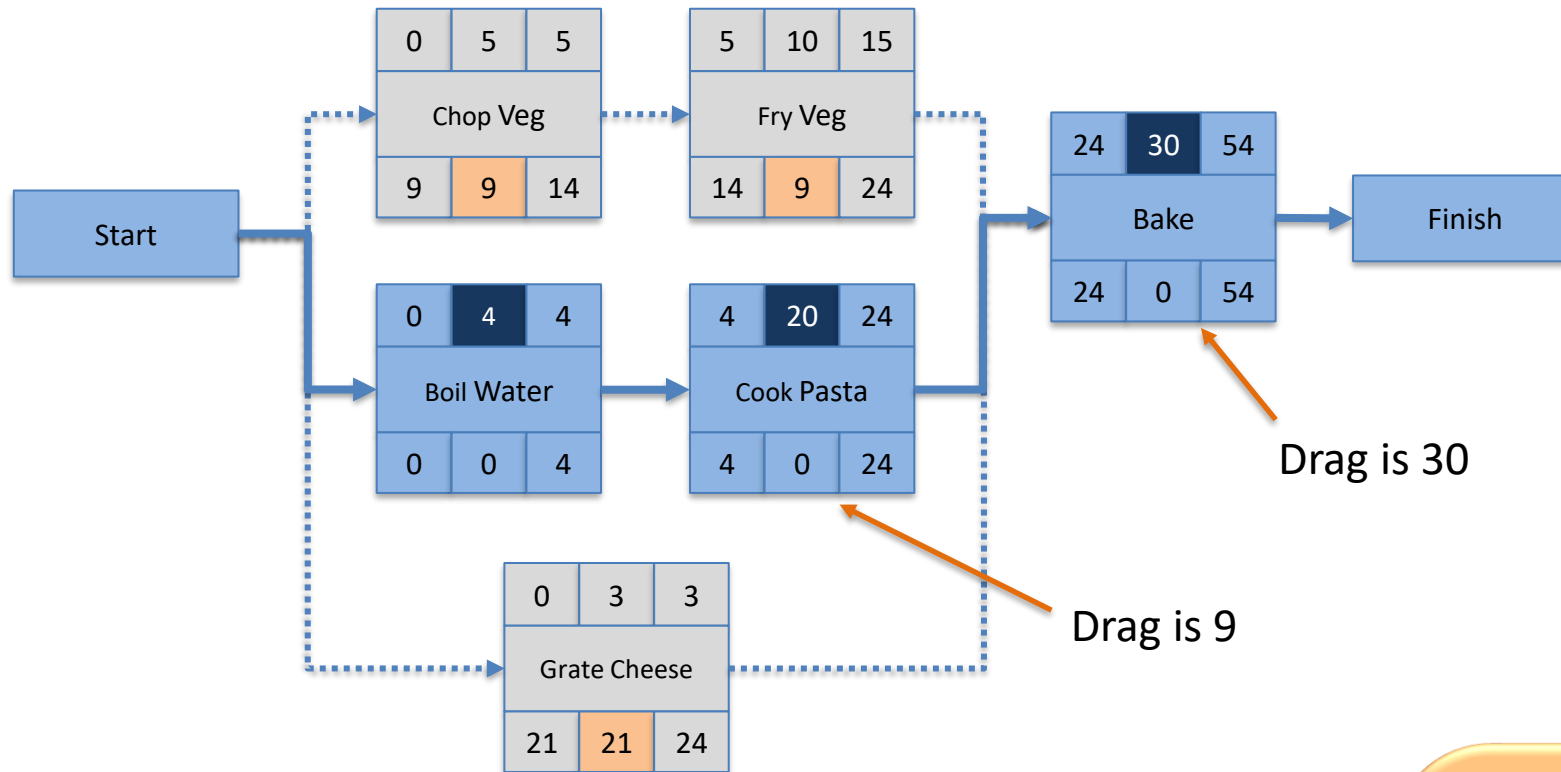
ES	D	EF
Activity		
LS	TF	LF

Drag

Amount of time a critical task delays the project.

$$= \min(D , \min(TF_{\text{parallel}}))$$

CPM: Drag Time



Key:

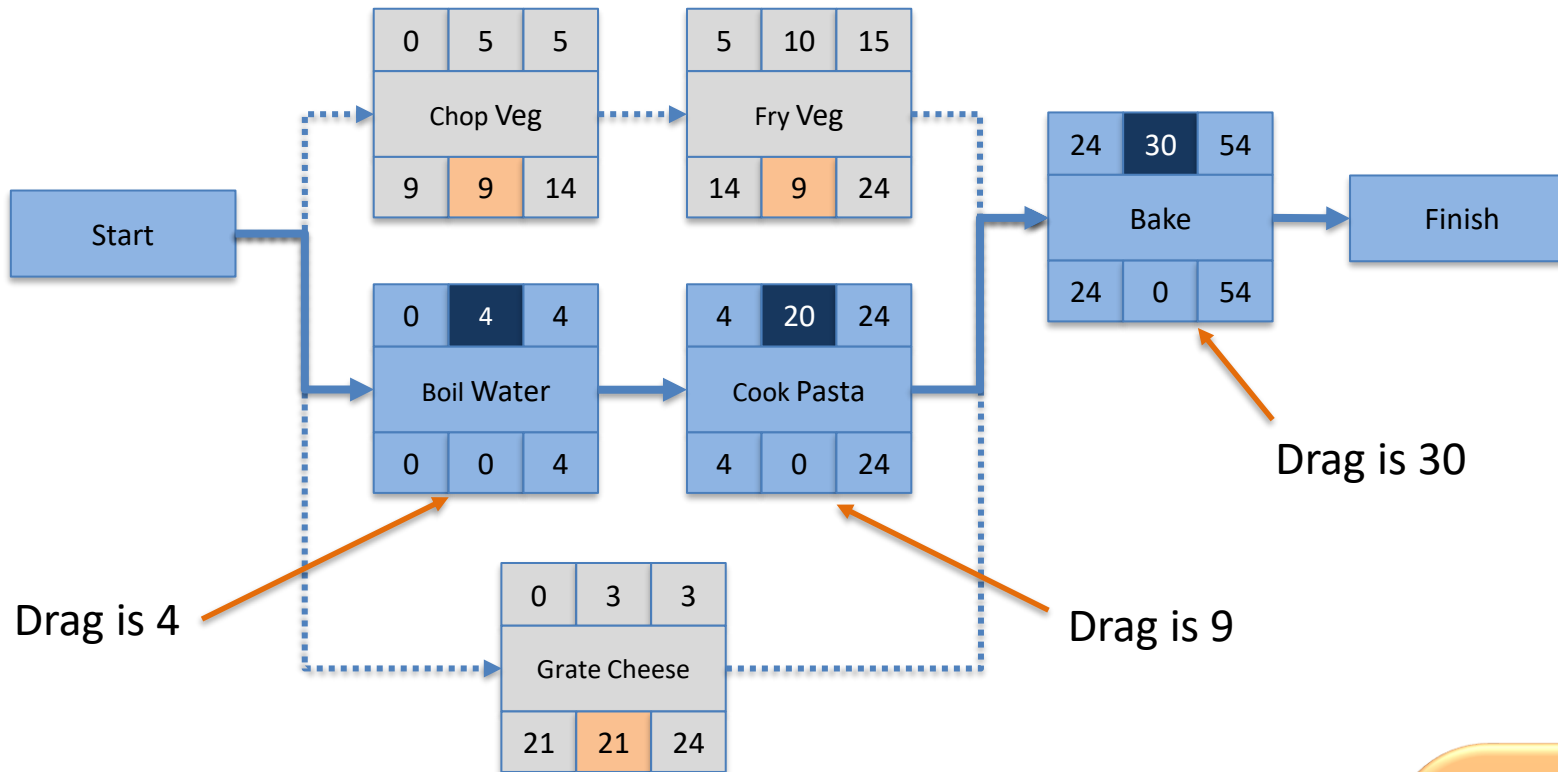
ES	D	EF
Activity		
LS	TF	LF

Drag

Amount of time a critical task delays the project.

$$= \min(D , \min(TF_{\text{parallel}}))$$

CPM: Drag Time



Key:

ES	D	EF
Activity		
LS	TF	LF

Drag

Amount of time a critical task delays the project.

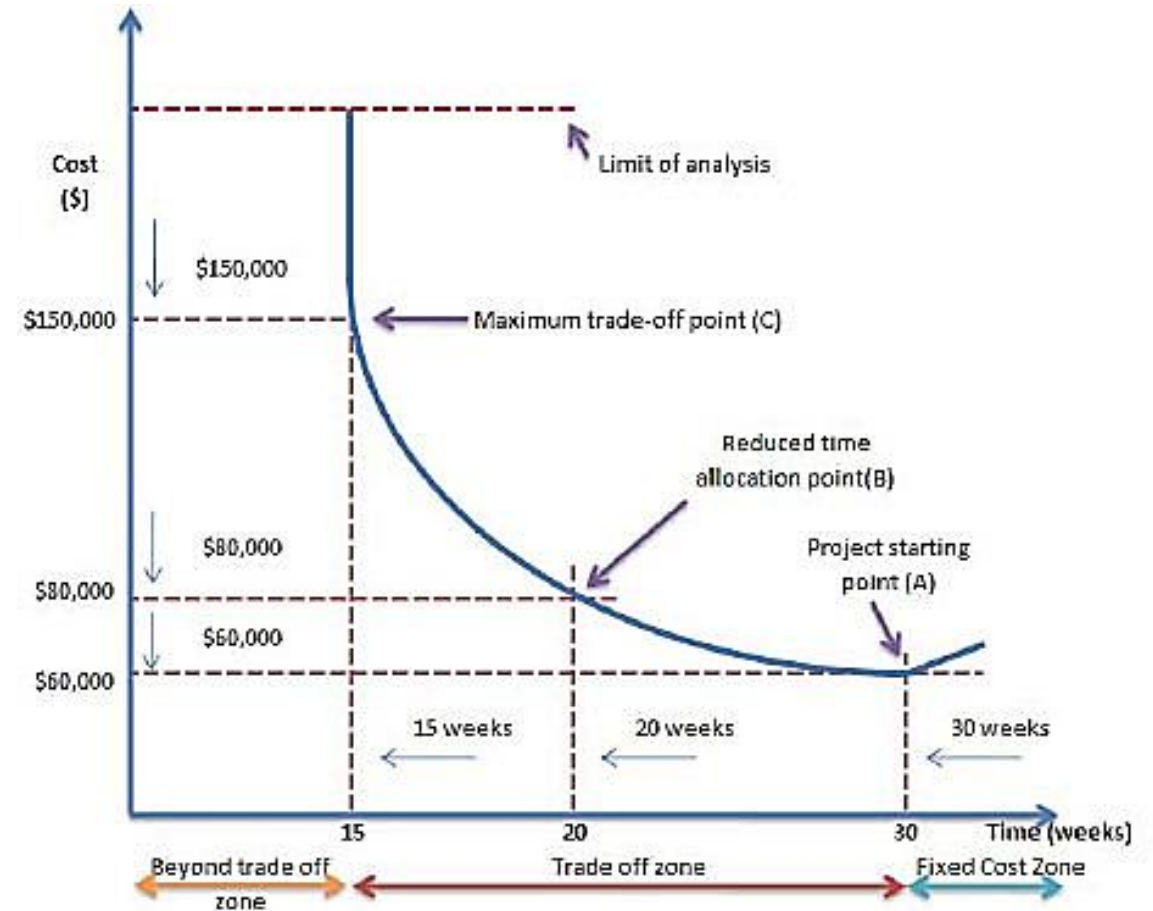
$$= \min(D , \min(TF_{\text{parallel}}))$$

CPM: Crashing

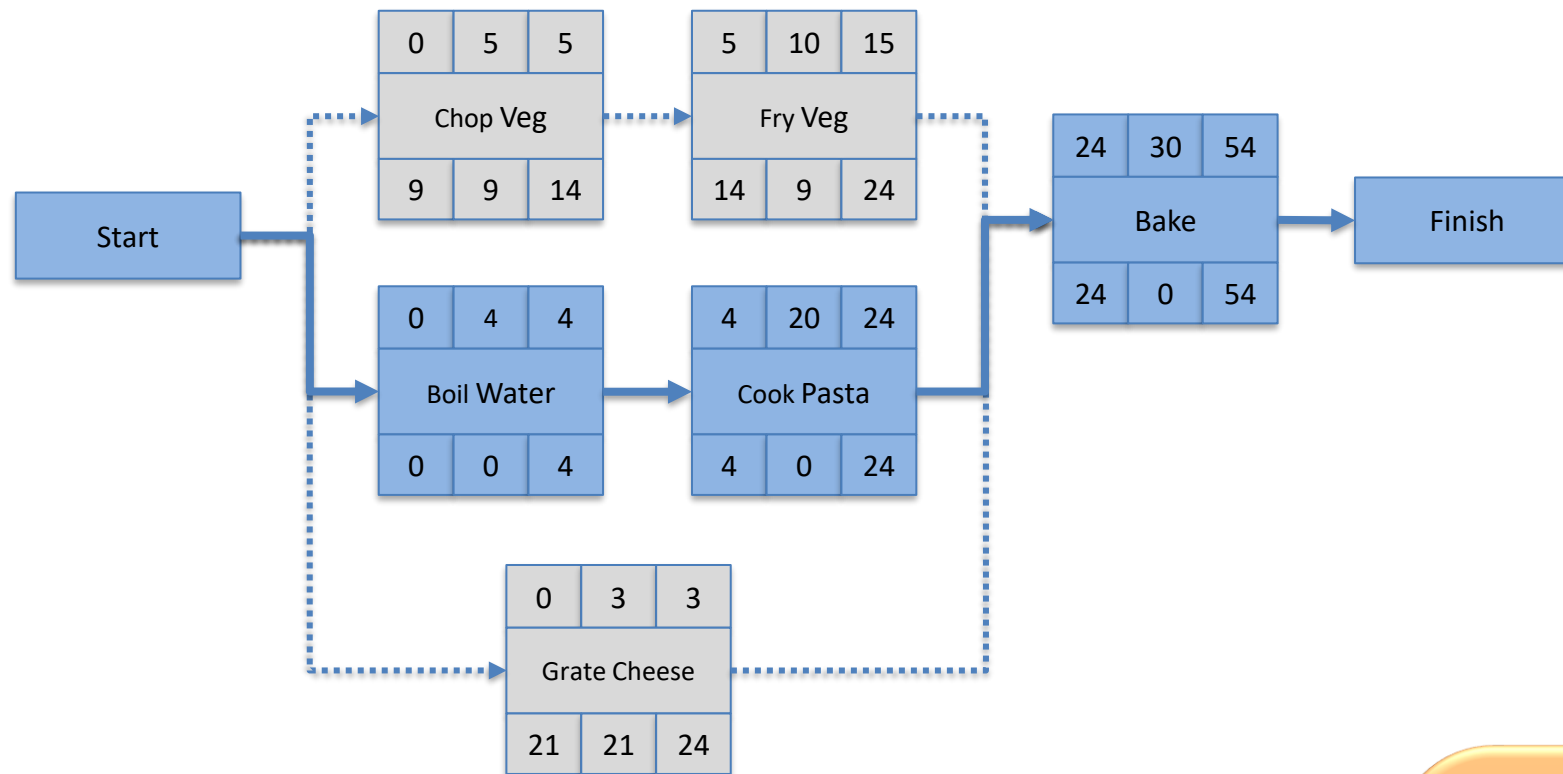
Crashing

“A schedule compression technique in which costs and schedule trade-offs are analysed to determine how to obtain the greatest amount of compression for the least incremental cost.” - *PMBOK® Guide*

Crash Duration: shortest possible time for which an activity can be scheduled, to speed up the whole project
= D – Drag



CPM: Crashing



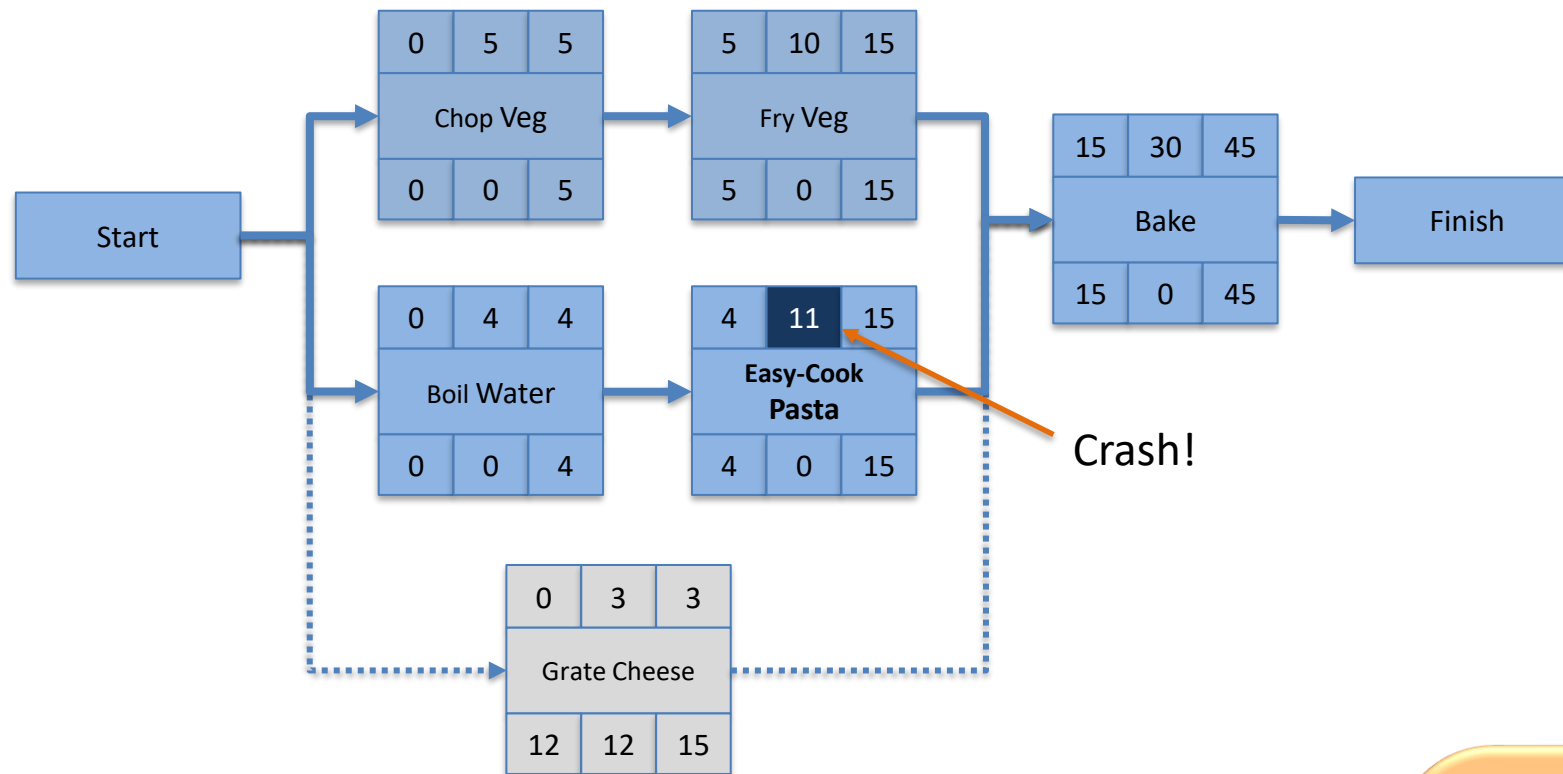
Key:

ES	D	EF
Activity		
LS	TF	LF

Crash

Reduce duration of tasks on critical path

CPM: Crashing



Key:

ES	D	EF
Activity		
LS	TF	LF

Crash

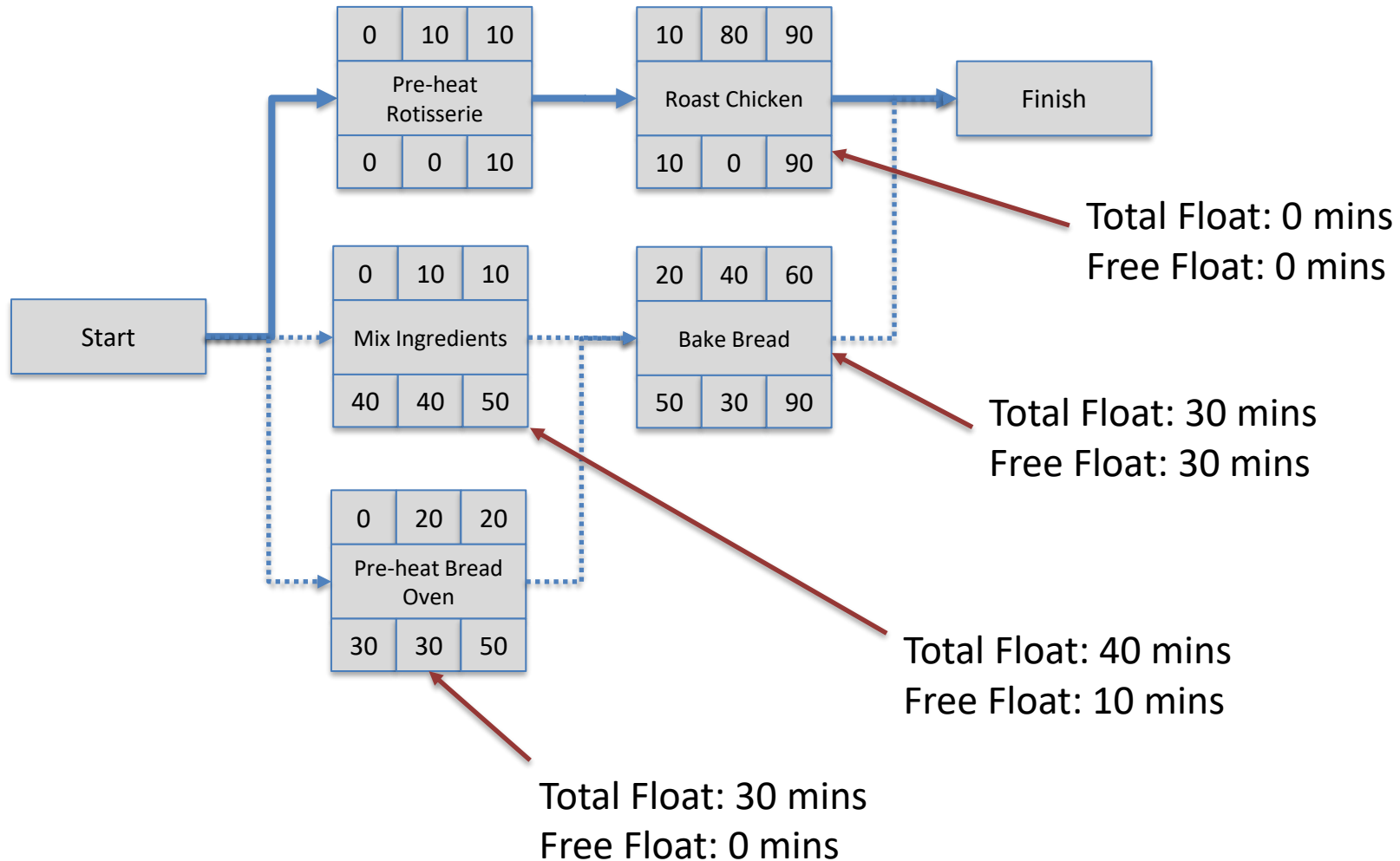
Reduce duration of tasks on critical path

CPM: Float Time

Float Time

- The critical path has a *Total Float (Slack)* of 0
- An activity can be delayed by its float time without causing other delays
- **Total Float (TF):**
The total amount of time that an activity may be delayed from its early start date without delaying the **project finish date**.
$$= LF - EF = LS - ES$$
- **Free Float (FF):**
The amount of time that an activity can be delayed without delaying the early start date of any **subsequent activities**
$$= ES_{\text{next}} - EF$$
- $FF \leq TF$

CPM: Float Time

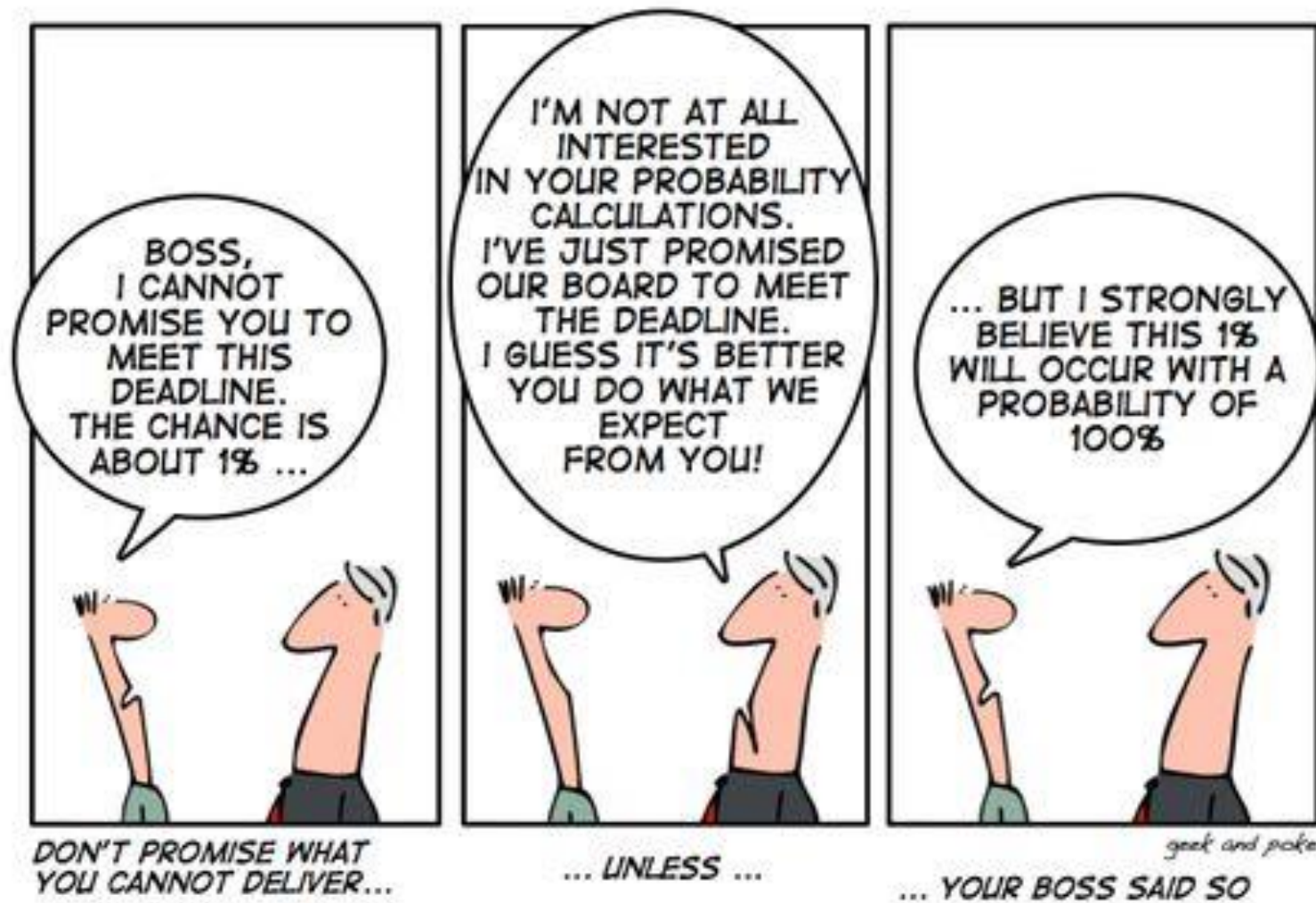


Summary

- CPM finds the critical path from a PND
- Forward pass propagates **durations** subject to **dependencies**
- Backward pass calculates the **float time (slack)** in the schedule
- Conversely, **critical** activities **drag** the project, and are the target of **crashing**

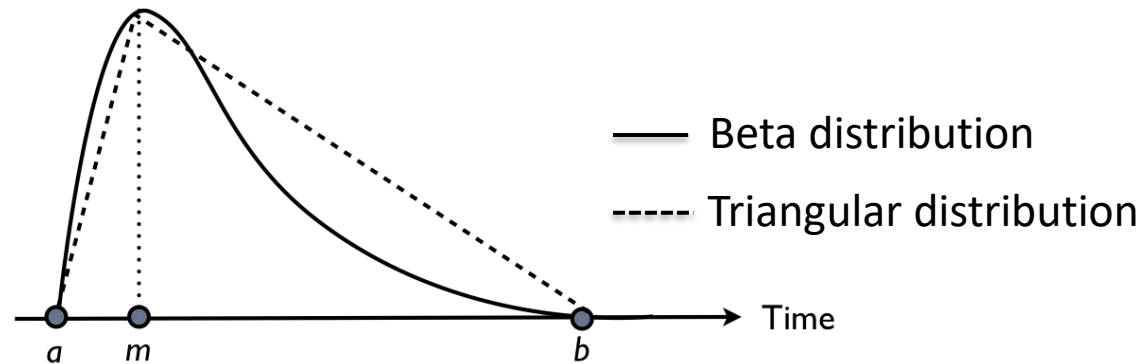
Today

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - **Program Evaluation and Review Technique (PERT)**



Program Evaluation and Review Technique (PERT)

- Takes skeptical view of time estimates
- Variation on CPM using **Three-point** estimation:
 - Shortest possible time each activity will take (a)
 - Most likely length of time (m)
 - Longest time, if the activity takes longer than expected (b)



Program Evaluation and Review Technique (PERT)

———— Beta distribution

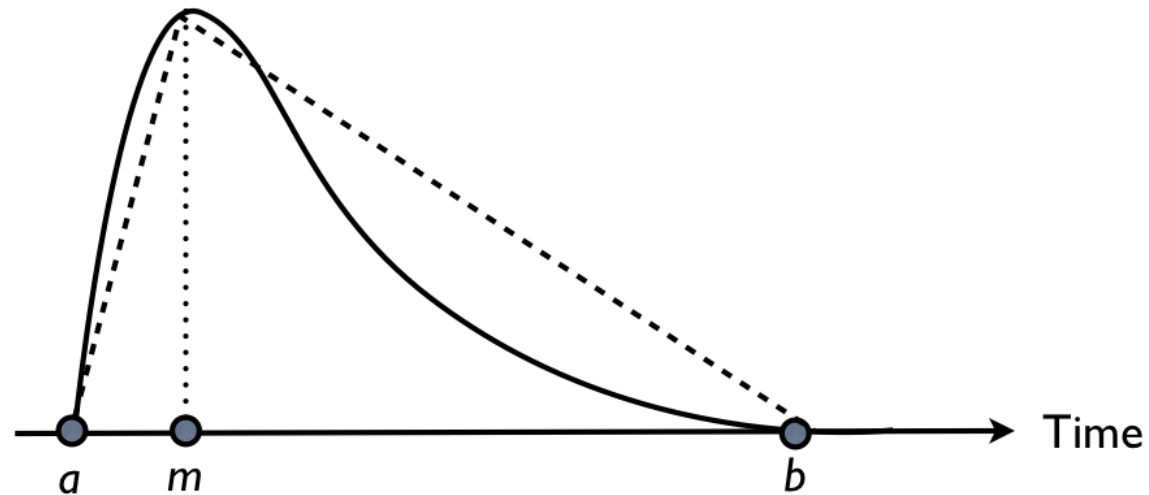
$$\text{Mean } t = \frac{a + 4m + b}{6}$$

$$\text{Variance } \sigma^2 = \left(\frac{b - a}{6}\right)^2$$

----- Triangular distribution

$$t = \frac{a + b + m}{3}$$

$$\sigma^2 = \frac{a^2 + b^2 + m^2 - ab - am - bm}{18}$$

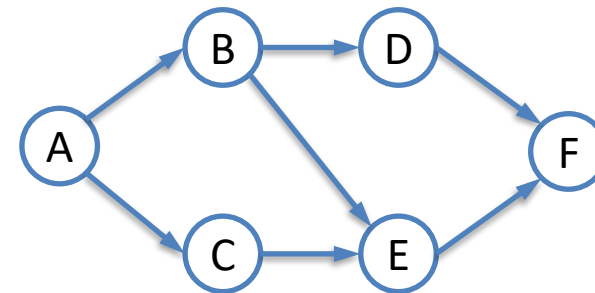


CPM and PERT

Task	Predecessors
A	-
B	A
C	A
D	B
E	B, C
F	D, E

CPM and PERT

Task	Predecessors
A	-
B	A
C	A
D	B
E	B, C
F	D, E

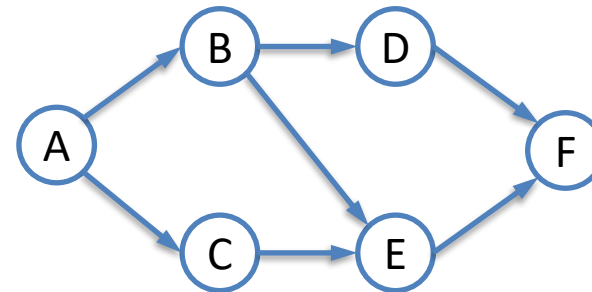


CPM

Fixed Duration

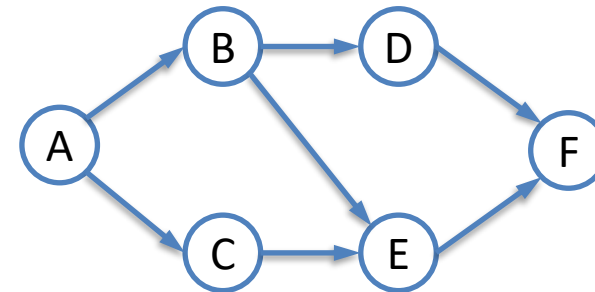


Task	Predecessors	m
A	-	5
B	A	2
C	A	0.5
D	B	1
E	B, C	2
F	D, E	3



PERT

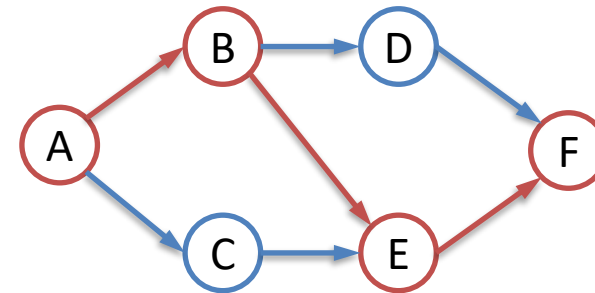
		Optimistic	Realistic	Pessimistic
		↓	↓	↓
Task	Predecessors	a	m	b
A	-	2	5	8
B	A	1	2	9
C	A	0.25	0.5	3.75
D	B	1	1	7
E	B, C	1	2	9
F	D, E	1	3	11



PERT

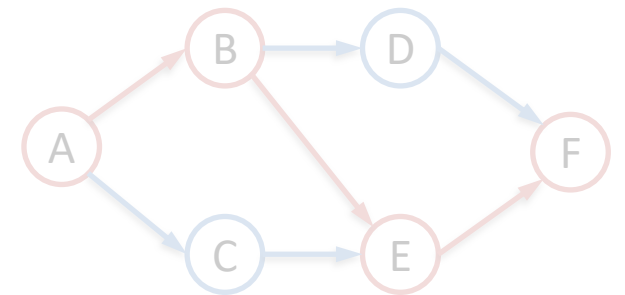
Task	Predecessors	Optimistic	Realistic	Pessimistic
		a	m	b
A	-	2	5	8
B	A	1	2	9
C	A	0.25	0.5	3.75
D	B	1	1	7
E	B, C	1	2	9
F	D, E	1	3	11

Use CPM!

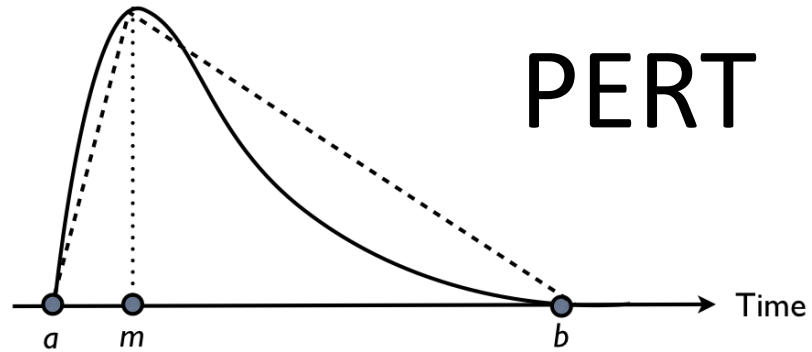


PERT

		Optimistic	Realistic	Pessimistic
		↓	↓	↓
Task	Predecessors	a	m	b
A	-	2	5	8
B	A	1	2	9
C	A	0.25	0.5	3.75
D	B	1	1	7
E	B, C	1	2	9
F	D, E	1	3	11

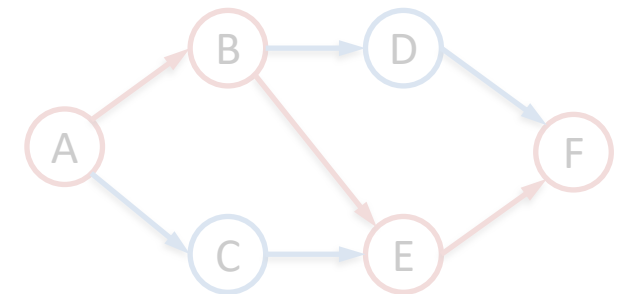


PERT

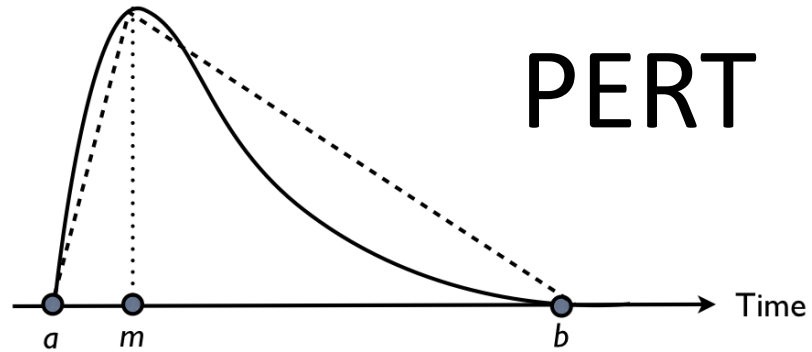


Optimistic Realistic Pessimistic

Task	Predecessors	<i>a</i>	<i>m</i>	<i>b</i>
A	-	2	5	8
B	A	1	2	9
C	A	0.25	0.5	3.75
D	B	1	1	7
E	B, C	1	2	9
F	D, E	1	3	11



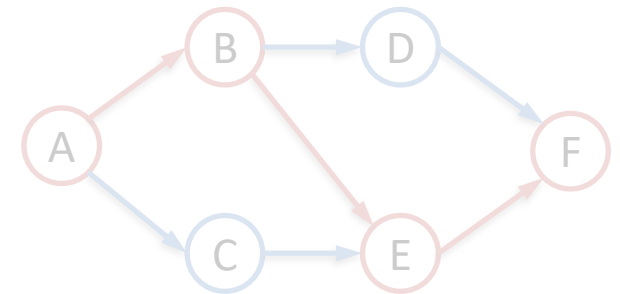
PERT



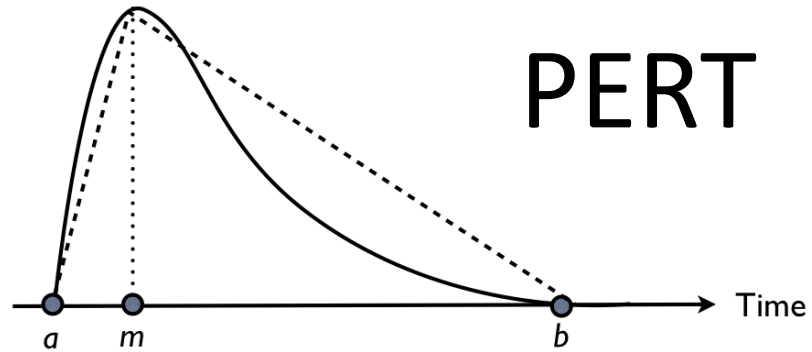
$$\text{Mean } t = \frac{a + 4m + b}{6}$$

$$\text{Variance } \sigma^2 = \left(\frac{b - a}{6}\right)^2$$

Task	Predecessors	a	m	b	t	σ
A	-	2	5	8	5	1
B	A	1	2	9	3	8/6
C	A	0.25	0.5	3.75	1	3.5/6
D	B	1	1	7	2	1
E	B, C	1	2	9	3	8/6
F	D, E	1	3	11	4	10/6



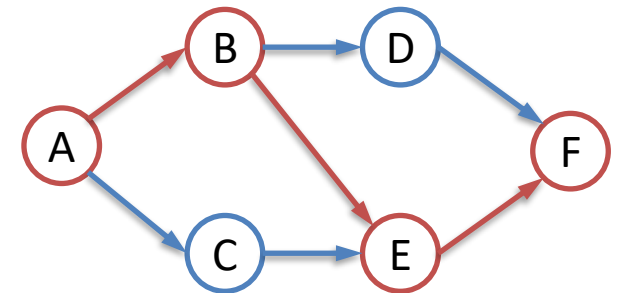
PERT



$$\text{Mean } t = \frac{a + 4m + b}{6}$$

$$\text{Variance } \sigma^2 = \left(\frac{b - a}{6}\right)^2$$

Task	Predecessors	<i>a</i>	<i>m</i>	<i>b</i>	<i>t</i>	σ
A	-	2	5	8	5	1
B	A	1	2	9	3	8/6
C	A	0.25	0.5	3.75	1	3.5/6
D	B	1	1	7	2	1
E	B, C	1	2	9	3	8/6
F	D, E	1	3	11	4	10/6

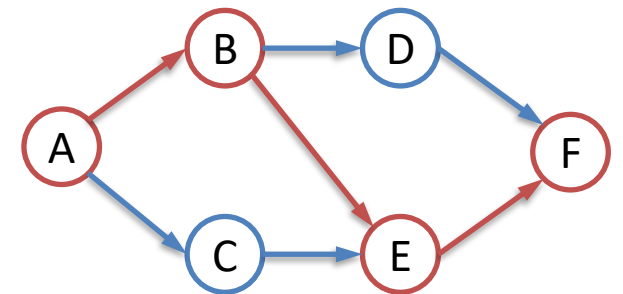


PERT

Task	Predecessors	a	m	b	t	σ
A	-	2	5	8	5	1
B	A	1	2	9	3	8/6
C	A	0.25	0.5	3.75	1	3.5/6
D	B	1	1	7	2	1
E	B, C	1	2	9	3	8/6
F	D, E	1	3	11	4	10/6

$$E(t) = 5 + 3 + 3 + 4 = 15$$

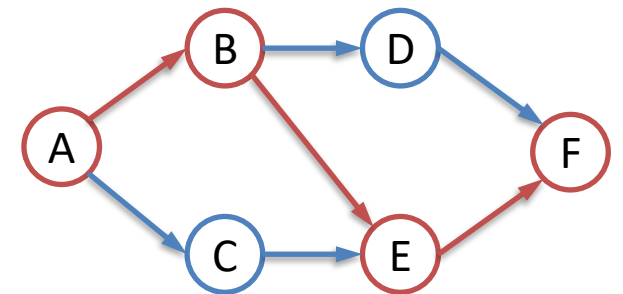
$$E(\sigma^2) = 1^2 + (8/6)^2 + (8/6)^2 + (10/6)^2 = 7.33$$



PERT

Task	Predecessors	a	m	b	t	σ
A	-	2	5	8	5	1
B	A	1	2	9	3	8/6
C	A	0.25	0.5	3.75	1	3.5/6
D	B	1	1	7	2	1
E	B, C	1	2	9	3	8/6
F	D, E	1	3	11	4	10/6

$$E(t) = 15$$
$$E(\sigma^2) = 7.33$$



PERT

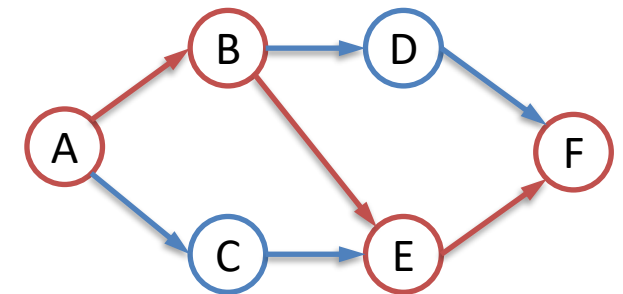
Task	Predecessors	a	m	b	t	σ
A	-	2	5	8	5	1
B	A	1	2	9	3	8/6
C	A	0.25	0.5	3.75	1	3.5/6
D	B	1	1	7	2	1
E	B, C	1	2	9	3	8/6
F	D, E	1	3	11	4	10/6

$$E(t) = 15$$

$$E(\sigma^2) = 7.33$$

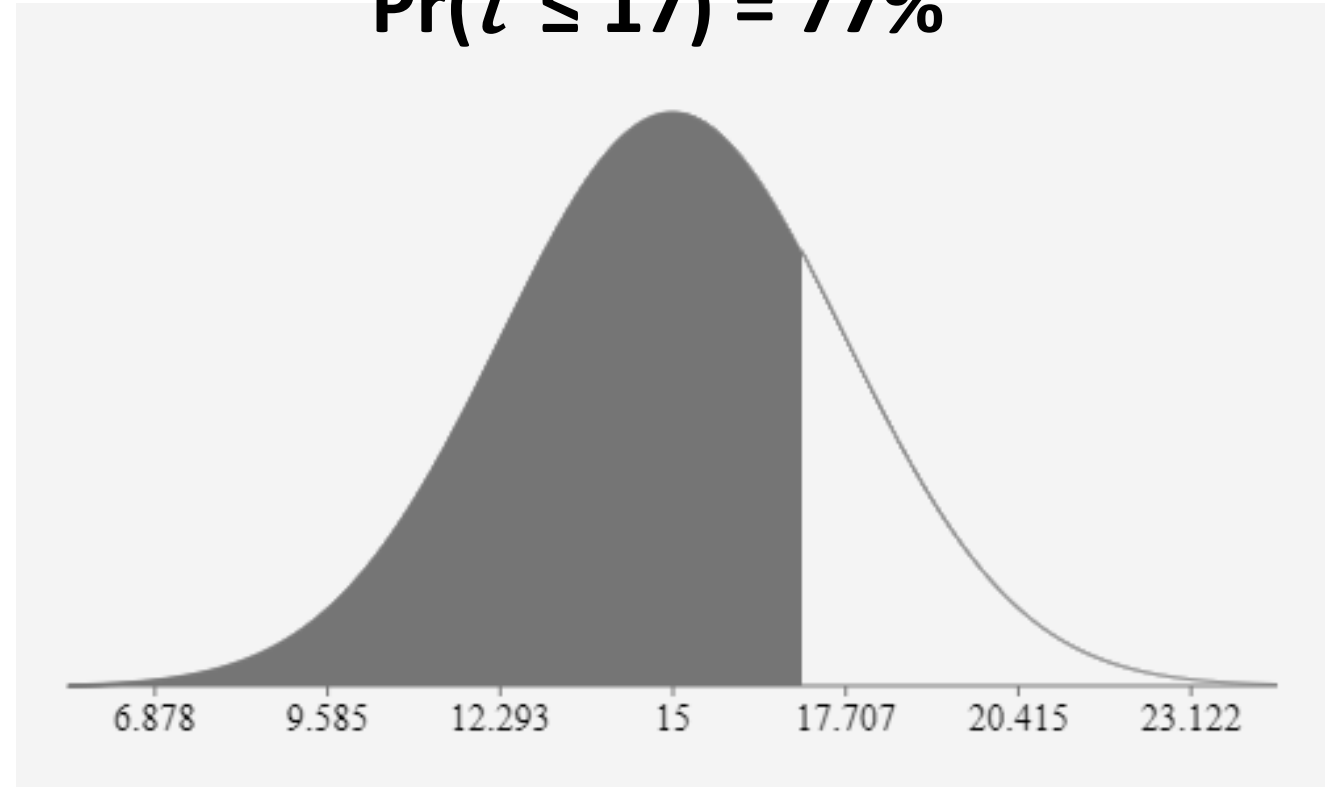
Assume Normal Distribution:

$$\text{Mean}=15, \text{SD}= \sqrt{7.33} = 2.7$$



PERT

$$\Pr(t \leq 17) = 77\%$$



$$E(t) = 15$$
$$E(\sigma^2) = 7.33$$

Assume Normal Distribution:

$$\text{Mean}=15, \text{SD}= \sqrt{7.33} = 2.7$$

Summary

- PERT
 - like CPM but with **uncertainty**.
 - Uses a **probability distribution**, not a fixed estimate.
 - Allows us to make statistical inferences (not promises we are likely to break).
- But
 - Makes assumptions of normality,
 - Requires uncertainty estimation,
 - Gives a false sense of precision

Recap

- Planning is a continuous process
- Scope Management: Work Breakdown Structure (WBS)
 - Work packages
 - Deliverable-oriented (vs Objective- / Process- oriented)
- Time Management
 - Estimation
 - Dependencies and Gantt Charts
 - Project Network Diagram
 - Critical Path Method (CPM)
 - Program Evaluation and Review Technique (PERT)