System Analysis and Design

Project Planning

Prof. Rania Elgohary
rania.elgohary@cis.asu.edu.eg
Dr. Yasmine Afify
yasmine.afify@cis.asu.edu.eg

Project planning

- **Project planning** involves breaking down the work into parts and assign these to project team members, anticipate problems that might arise and prepare tentative solutions to those problems.
- The **project plan**, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.

Planning stages

- At the **proposal stage**, when you are bidding for a contract to develop or provide a software system.
- During the **project startup** phase, when you have to plan who will work on the project, how the project will be broken down into increments, how resources will be allocated across your company, etc.
- Periodically **throughout the project**, when you modify your plan in the light of experience gained and information from monitoring the progress of the work.

Proposal planning

- ✧ Planning may be necessary with only outline software requirements.
- ✧ The aim of planning at this stage is to provide information that will be used in setting a price for the system to customers.
- ✧ Project pricing involves estimating how much the software will cost to develop, taking into account factors such as staff costs, hardware costs, software costs, etc..

Project startup planning

- At this stage, you know more about the system requirements but do not have design or implementation information
- Create a plan with enough detail to make decisions about the project budget and staffing (this plan is the basis for project resource allocation)
- The startup plan should also define project monitoring mechanisms
- A startup plan is still needed for agile development to allow resources to be allocated to the project

Development planning

- The project plan should be regularly **amended** as the project progresses and you know more about the software and its development
- The project schedule, cost-estimate and risks have to be regularly **revised**

Software Pricing

Software pricing

- ✧ Estimates are made to discover the cost, to the developer, of producing a software system.
  - ▪ You take into account, hardware, software, travel, training and effort costs.
- ✧ There is not a simple relationship between the development cost and the price charged to the customer.
- ✧ Broader organisational, economic, political and business considerations influence the price charged.

Factors affecting software pricing

Factor	Description
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Cost estimate uncertainty	If an organization is unsure of its cost estimate, it may increase its price by a contingency over and above its normal profit.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a smaller than normal profit or break even than to go out of business. Cash flow is more important than profit in difficult economic times.

Factors affecting software pricing

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the organization the opportunity to make a greater profit later. The experience gained may also help it develop new products.
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract is awarded, high prices can be charged for changes to the requirements.

Pricing strategies

- ✧ Under pricing
  - A company may underprice a system in order to gain a contract that allows them to retain staff for future opportunities
  - A company may underprice a system to gain access to a new market area
- ✧ Increased pricing
  - The price may be increased when a buyer wishes a fixed-price contract and so the seller increases the price to allow for unexpected risks

Plan-driven Development

Plan-driven development

- Plan-driven development is an approach to software engineering where the development process is planned in detail.
- Plan-driven development is the 'traditional' way of managing large software development projects.
- A **project plan** is created that records the work to be done, who will do it, the development schedule and the work products.
- Managers use the plan to support project decision making and as a way of measuring progress.

Plan-driven development – pros and cons

- The arguments **in favor of** a plan-driven approach are that early planning allows organizational issues (availability of staff, other projects, etc.) to be closely taken into account, and that potential problems and dependencies are discovered before the project starts, rather than once the project is underway.
- The principal argument **against** plan-driven development is that many early decisions have to be revised because of changes to the environment in which the software is to be developed and used.

Project plans

- In a plan-driven development project, a project plan sets out the resources available to the project, the work breakdown and a schedule for carrying out the work.
- Plan sections
  - Introduction
  - Project organization
  - Risk analysis
  - Hardware and software resource requirements
  - Work breakdown
  - Project schedule
  - Monitoring and reporting mechanisms

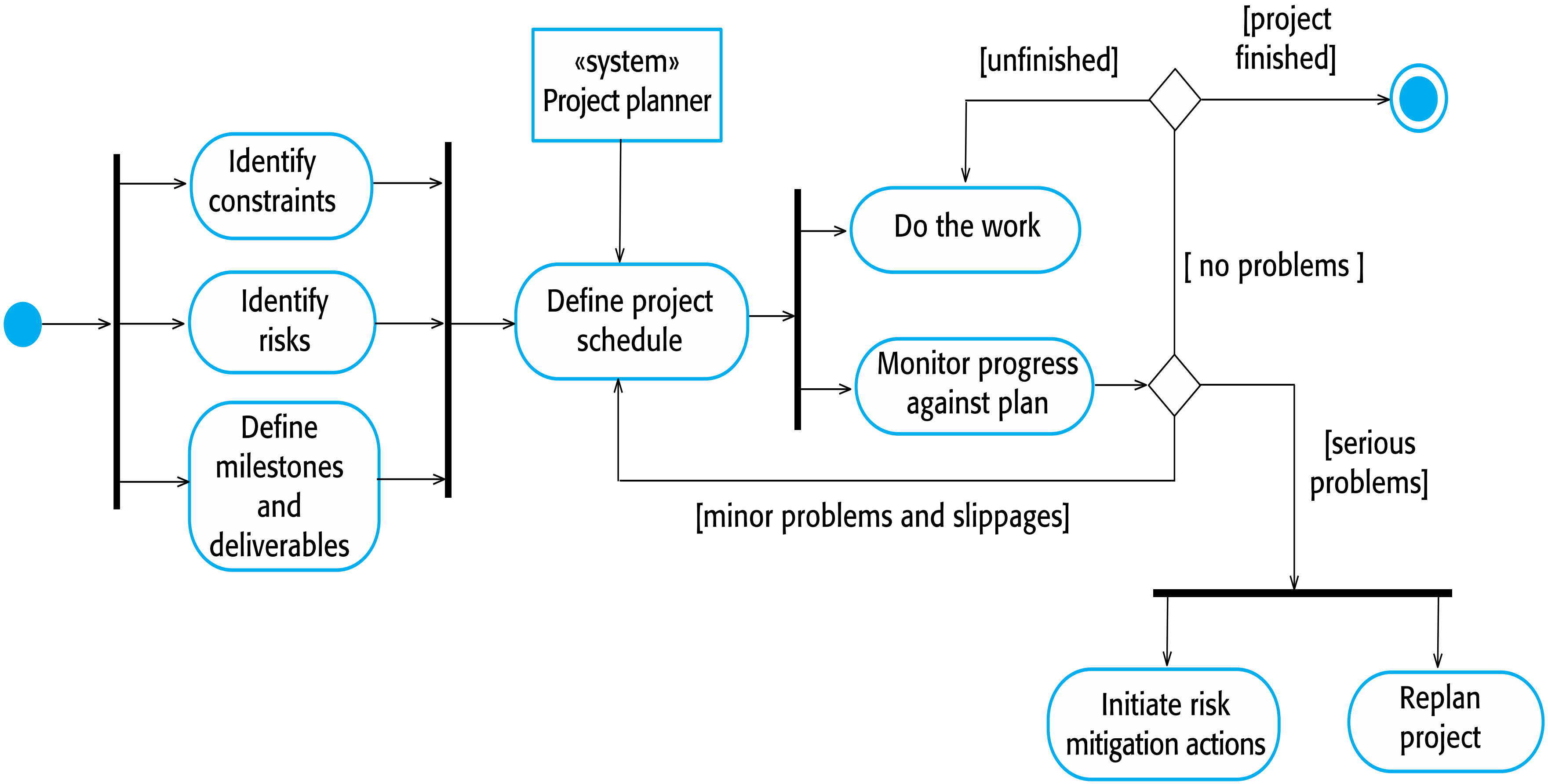
Project plan supplements

Plan	Description
Configuration management plan	Describes the configuration management procedures and structures to be used.
Deployment plan	Describes how the software and associated hardware (if any) will be deployed in the customer’s environment. This should include a plan for migrating data from existing systems.
Maintenance plan	Predicts the maintenance requirements, costs, and effort.
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources, and schedule used for system validation.

The planning process

- ✧ Project planning is an **iterative process** that starts when you create an initial project plan during the project startup phase.
- ✧ Plan **changes** are inevitable.
  - ▪ As more information about the system and the project team becomes available during the project, you should regularly revise the plan to reflect requirements, schedule and risk changes.
  - ▪ Changing business goals also leads to changes in project plans. As business goals change, this could affect all projects, which may then have to be re-planned.

The project planning process



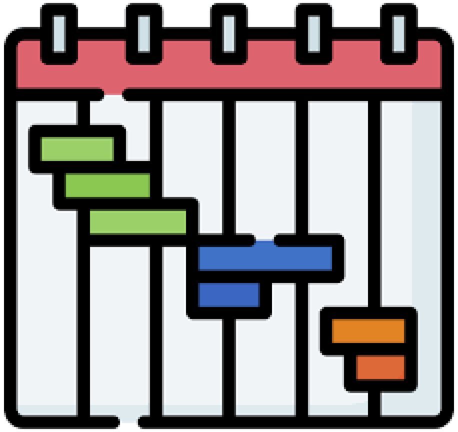
Planning assumptions

- You should make **realistic** rather than optimistic assumptions when you are defining a project plan.
- Problems always arise during a project, and these lead to project **delays**.
- Your initial assumptions and scheduling should therefore take unexpected problems into account.
- You should include **contingency** in your plan so that if things go wrong, then your delivery schedule is not seriously interrupted.

Risk mitigation

- If there are serious problems with the development work that are likely to lead to significant delays, you need to initiate **risk mitigation actions** to reduce the risks of project failure.
- In conjunction with these actions, you also have to **re-plan** the project.
- This may involve **renegotiating** the project constraints and deliverables with the customer.
- A **new schedule** of when work should be completed also has to be established and agreed with the customer.

Project Scheduling



Project scheduling

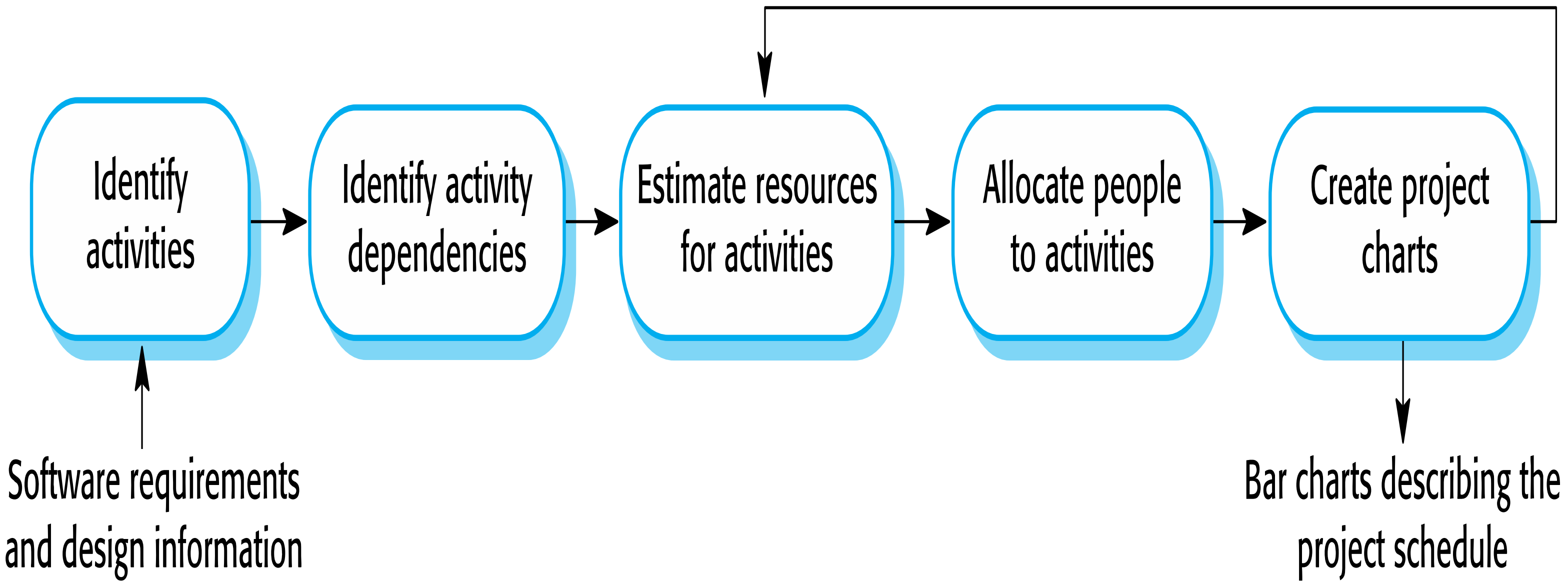
- Project scheduling is the process of deciding how the work in a project will be organized as **separate tasks**, and **when** and how these tasks will be executed.
- You **estimate** the **calendar time** needed to complete each task, the **effort** required, and **who** will work on the tasks that have been identified.
- You also estimate the **resources** needed to complete each task, such as the disk space required on a server, the time required on specialized hardware, such as a simulator, and what the travel budget will be.

Project scheduling activities

- Split project into tasks and estimate time and resources required to complete each task.
- Organize tasks concurrently to make optimal use of workforce.
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete.

Dependent on project managers intuition and experience.

The project scheduling process



Identify activities

Identify activity dependencies

Estimate resources for activities

Allocate people to activities

Create project charts

Software requirements and design information

Bar charts describing the project schedule

Scheduling problems

- ✧ Estimating the difficulty of problems and hence the cost of developing a solution is hard.
- ✧ Productivity is not proportional to the number of people working on a task.
- ✧ Adding people to a late project makes it later because of communication overheads.
- ✧ The unexpected always happens. Always allow contingency in planning.

Schedule presentation

- Graphical notations are normally used to illustrate the project schedule.
- These show the project breakdown into tasks. Tasks should not be too small. They should take about a week or two.
- Calendar-based
  - Bar charts are the most commonly used representation for project schedules. They show the schedule as activities or resources against time.
- Activity networks
  - Show task dependencies.

Project activities

- ✧ Project activities (tasks) are the basic planning element. Each activity has:
  - a duration in calendar days or months,
  - an effort estimate, which shows the number of person-days to complete the work,
  - a deadline by which the activity should be complete,
  - a defined end-point, which might be a document, the holding of a review meeting, the successful execution of all tests, etc.

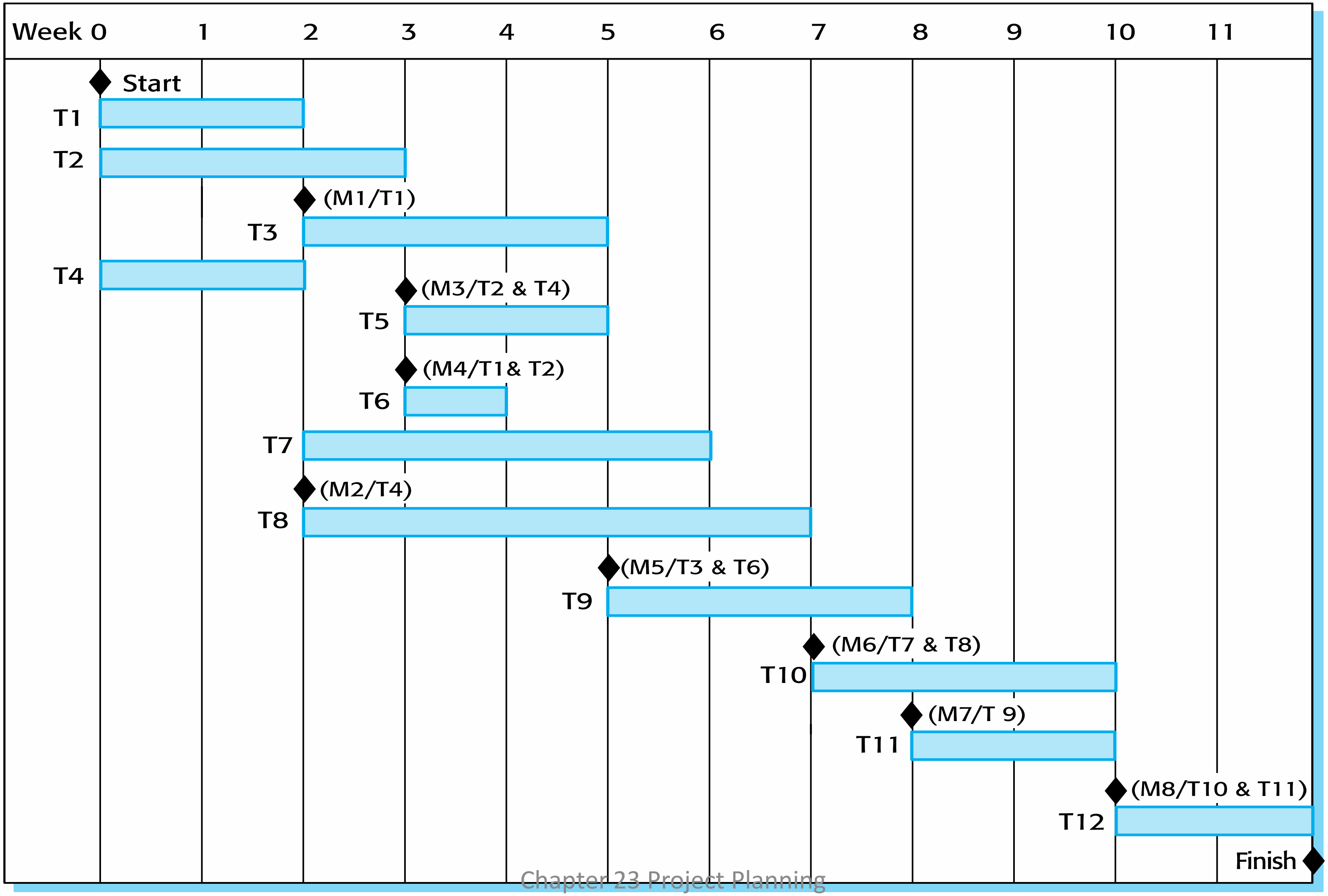
Milestones and deliverables

- **Milestones** are points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- **Deliverables** are work products that are delivered to the customer, e.g. a requirements document for the system.
- Normally, you should use a project planning tool, such as the Basecamp or Microsoft project, to create, update, and analyze project schedule information.
- Project management tools usually expect you to input project information into a table, and they create a database of project information. Bar charts and activity charts can then be generated automatically from this database.

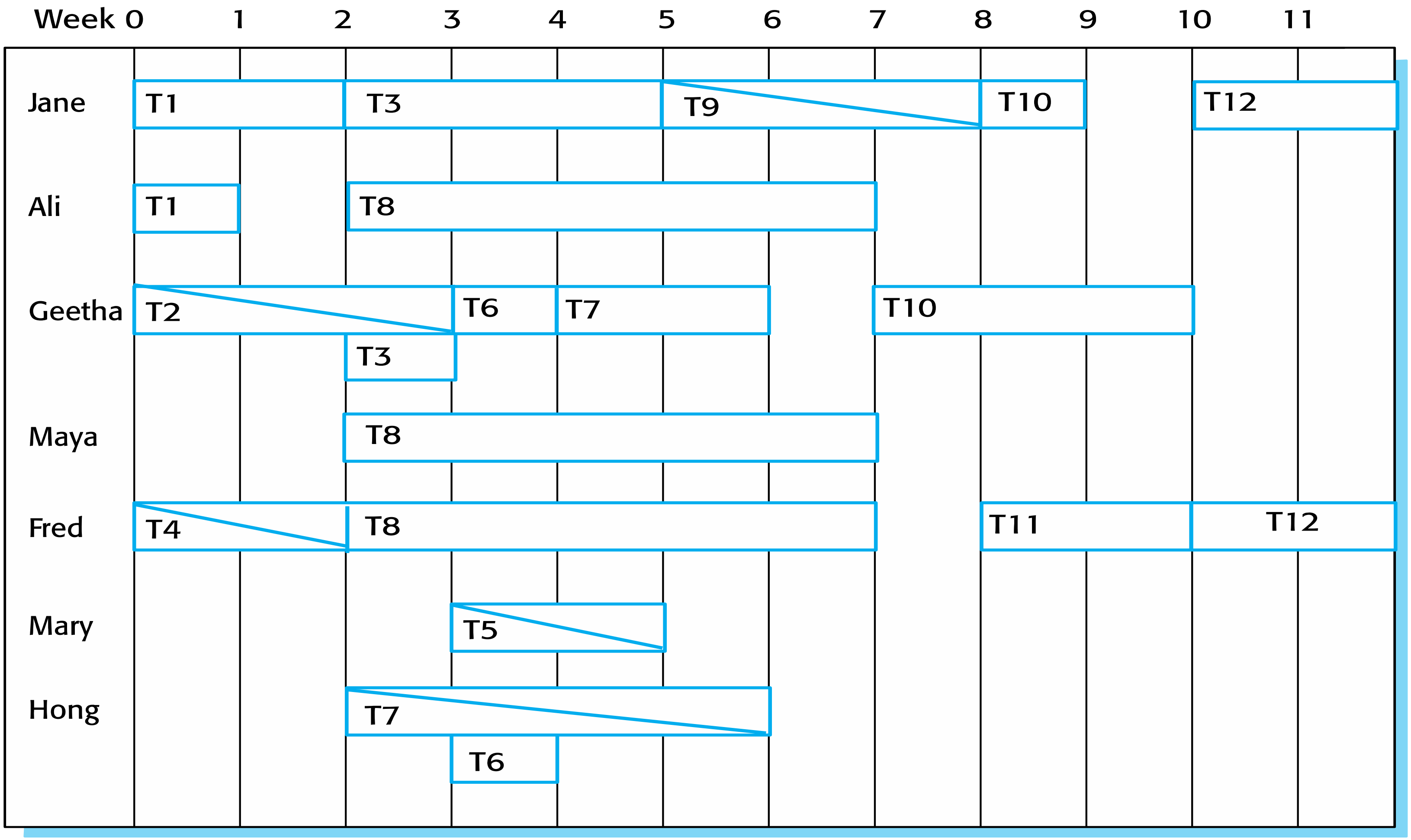
Tasks, durations, and dependencies

Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)

Activity bar chart



Staff allocation chart


Week 0 1 2 3 4 5 6 7 8 9 10 11

Jane T1 T3 T9 T10 T12

Ali T1 T8

Geetha T2 T3 T6 T7 T10

Maya T8

Fred T4 T8 T11 T12

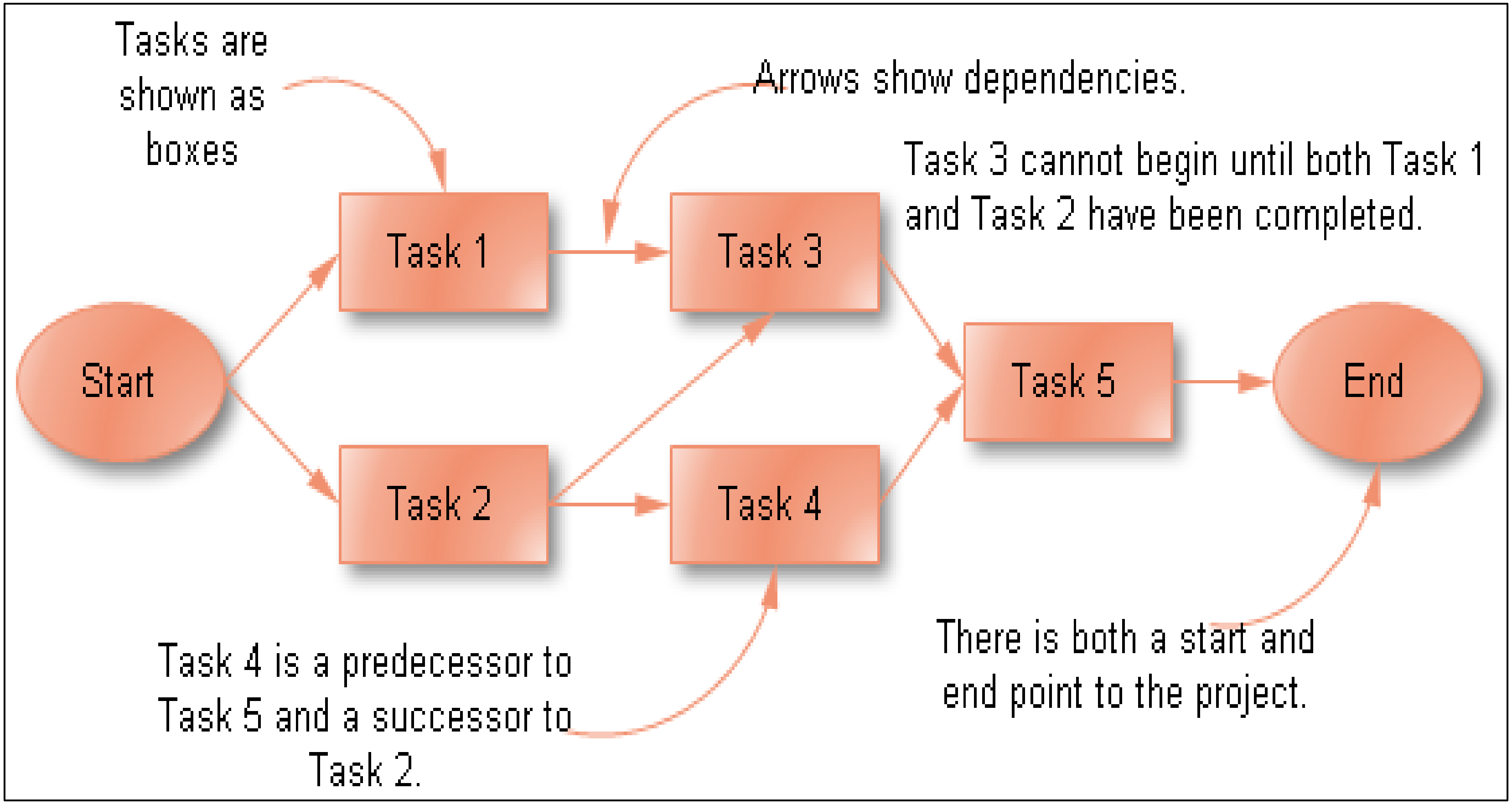
Mary T5

Hong T7 T6

Activity Network Diagram

- It is a diagram of project activities that shows the sequential **relationships of activities** using arrows and nodes.
- An activity network diagram tool is necessary for the identification of a **project's critical path** (which is used to determine the expected completion time of the project).

Activity Network Diagram



Tasks are shown as boxes

Arrows show dependencies.

Task 3 cannot begin until both Task 1 and Task 2 have been completed.

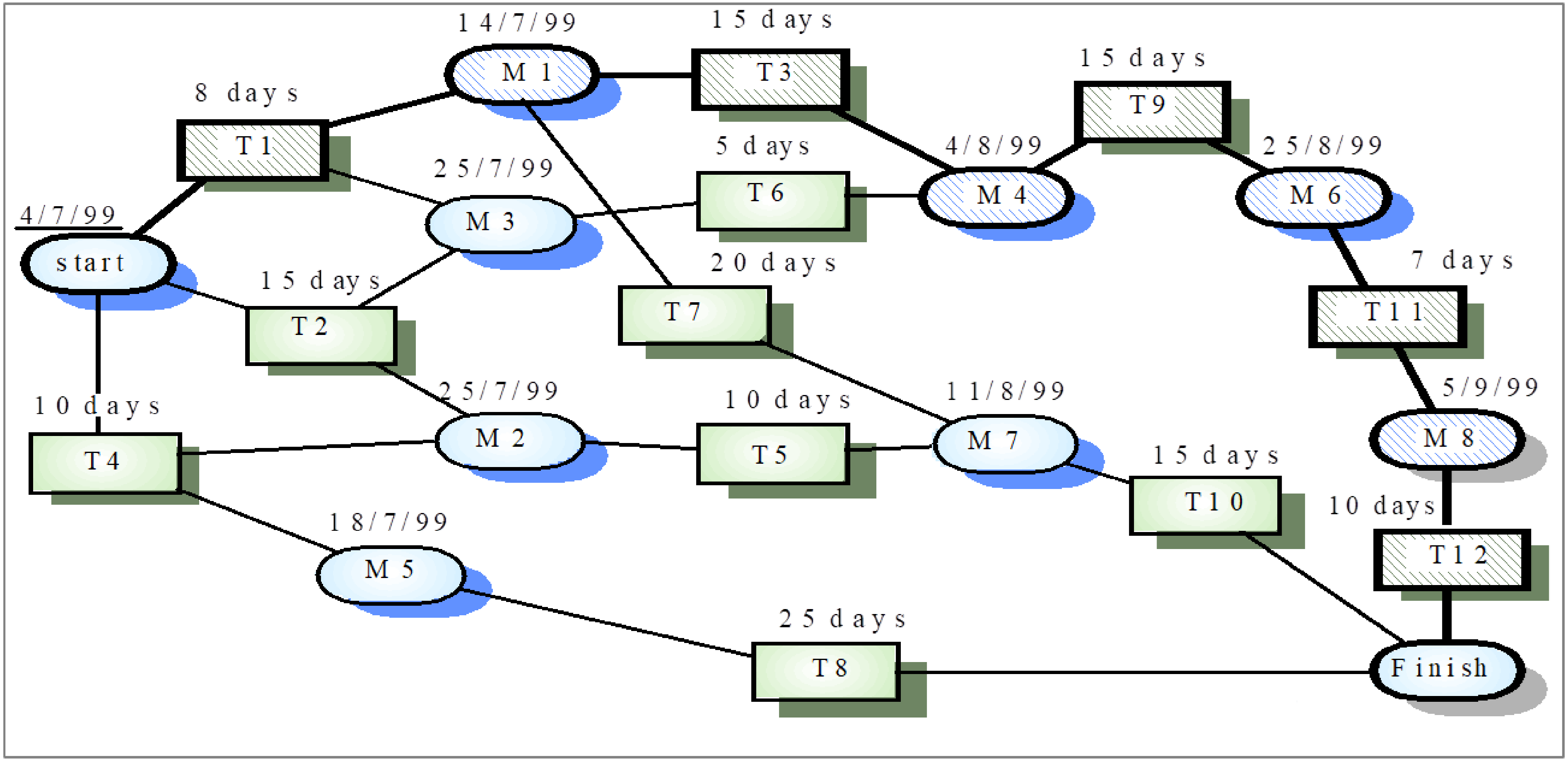
Task 4 is a predecessor to Task 5 and a successor to Task 2.

There is both a start and end point to the project.

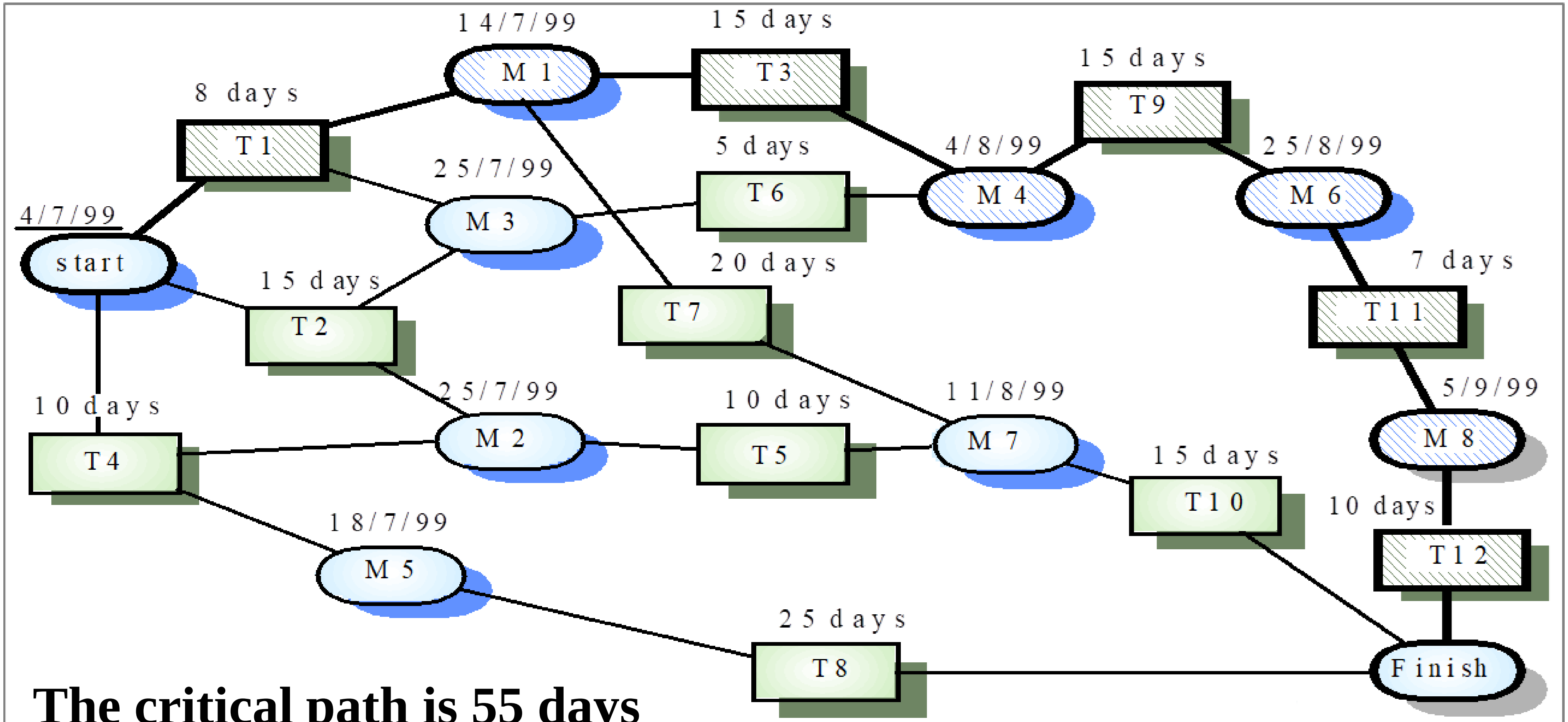
Tasks, durations, and dependencies

Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)

Activity Network



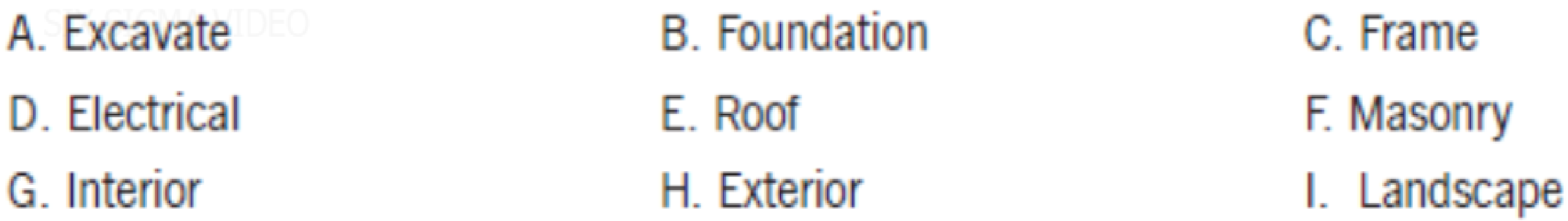
Activity Network



The critical path is 55 days

Example

Suppose the team is tasked with improving the process of building a house. The team lists the major steps: from the Excavate (digging) step through the landscaping step.



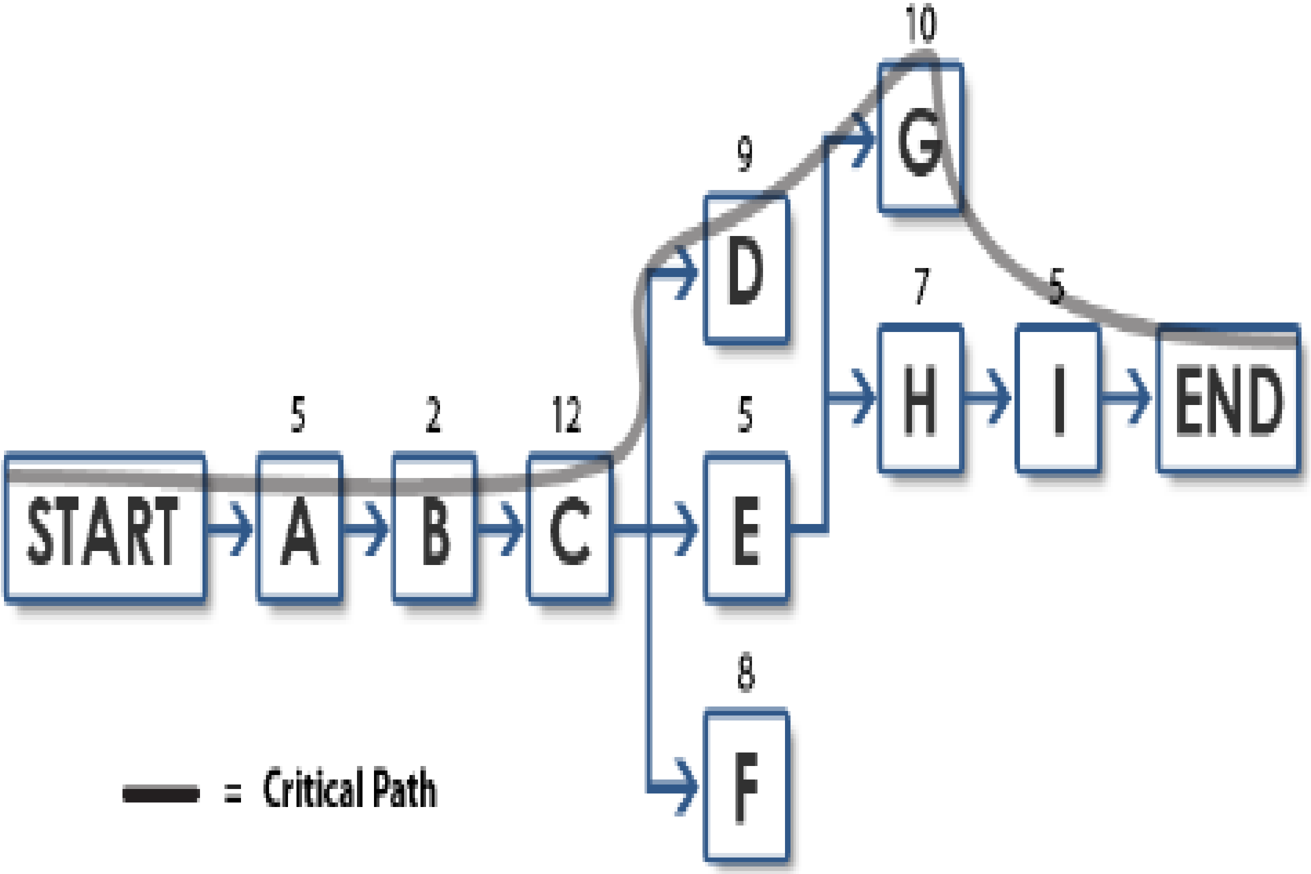
A. Excavate
B. Foundation
C. Frame
D. Electrical
E. Roof
F. Masonry
G. Interior
H. Exterior
I. Landscape

Example

- Some of the process steps (nodes A, B, and C) run in series, while other process steps (nodes D, E, and F) run in parallel.
- Notice that step B cannot happen until step A has been completed.
- Likewise, step C cannot happen until step B has completed.
- Step H cannot happen until steps D, E, and F have completed
- ALL need to be completed before Step H.
- This is important to know because those steps that are running in parallel most likely will have different expected completion times.

Example: Activity Network

A. Excavate	5 days
B. Foundation	2 days
C. Frame	12 days
D. Electrical	9 days
E. Roof	5 days
F. Masonry	8 days
G. Interior	10 days
H. Exterior	7 days
I. Landscape	5 days



Critical Path

- The critical path is a line that goes through all of the nodes that have the **longest expected completion times**.
- Nodes A, B, and C run in series, so the critical path is straightforward.
- Notice that between the three nodes that run in parallel (nodes D, E, and F), node D is expected to take the longest to complete as compared to the other two nodes. The critical path would run through nodes D and G because those particular nodes have the longest expected completion times.

Most Likely Time (Real Time)

By looking at the Activity Network Diagram, the team can easily see that the expected completion time as defined by the critical path is 50 days.

(5+2+12+9+10+7+5 = 50 days)

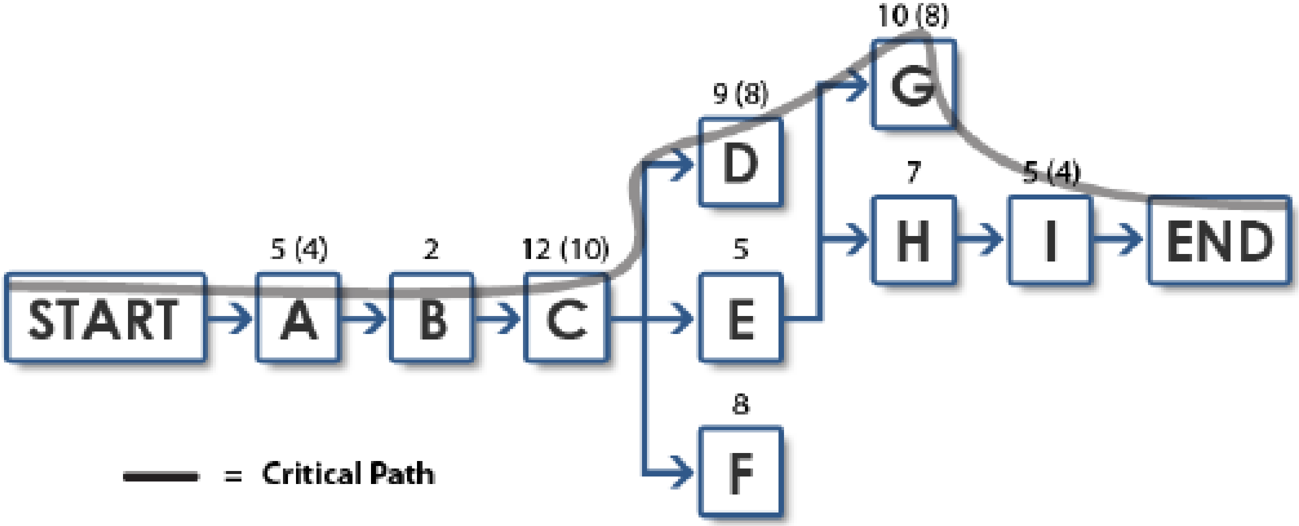
That's the **MOST LIKELY** time.

Optimistic Time

The team might want to know what the best case (Optimistic Time), in terms of time, would be.

To determine the best case, the critical path line must be followed.

To come up with that number, the team would decide upon the shortest possible time for each of the nodes, and then add those up. The numbers in parenthesis are the most optimistic times (4+2+10+8+8+7+4 = 43)

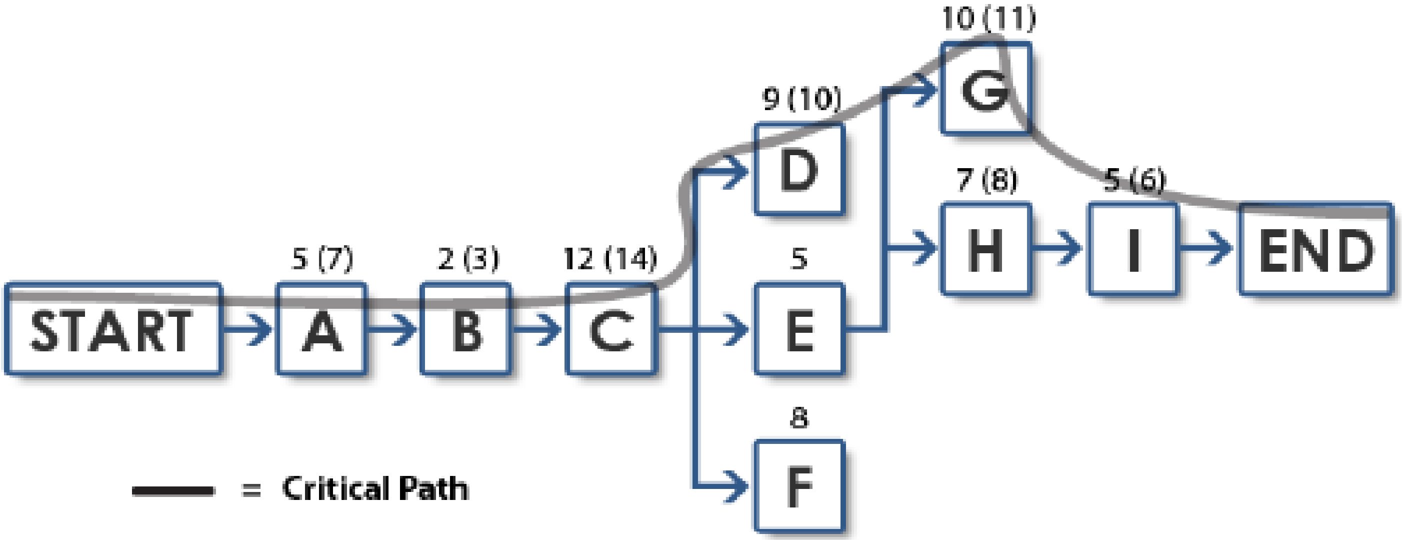


Pessimistic Time

The team also might want to know what the worst case (Pessimistic Time), in terms of time, would be.

To determine the worst case, the critical path line must be followed.

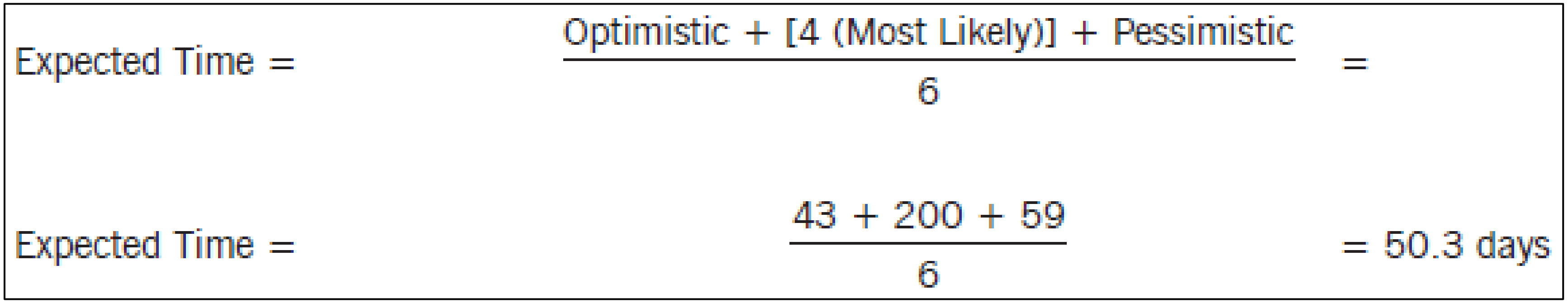
To come up with that number, the team would decide upon the longest possible time for each of the nodes, and then add those up. The numbers in parentheses are the most pessimistic times (7+3+14+10+11+8+6 = 59)



Expected Time

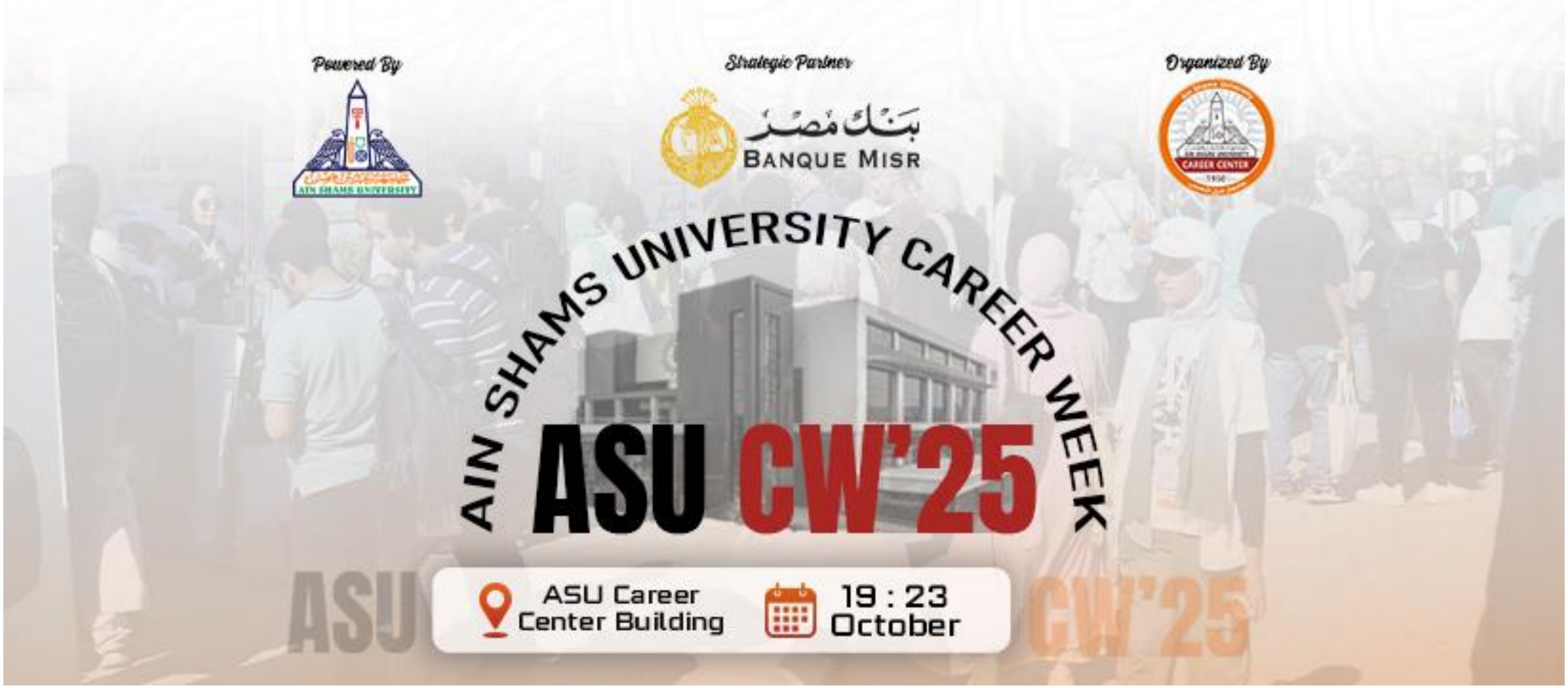
What does all of this mean?

It means the project most likely will take 50 days, but it could take 59 days, or it can be done as soon as 43 days.



$$\text{Expected Time} = \frac{\text{Optimistic} + [4\ (\text{Most Likely})] + \text{Pessimistic}}{6} =$$

$$\text{Expected Time} = \frac{43 + 200 + 59}{6} = 50.3 \text{ days}$$



.

1️⃣ Career Exhibition Days
📍 ASU Career Center – Students' Hostel, Abbassia
🕐 1 PM – 8 PM
Day 1 (Sun, Oct 19): Engineering, Computer Science

💡 Services: Career Talks – Mentorship Circle – Jobs & Internships – Networking
2️⃣ Open Recruitment Days
📍 ASU Career Center
🕘 9 AM – 4 PM
Day 4 (Wed, Oct 22): Engineering, Computer Science

Who can attend?

All ASU students & alumni can join Exhibition Days (according to faculty schedule).
To attend Recruitment Days, you must first join the Exhibition Days
& reserve your spot at the Career Center booth.

Register now: https://forms.gle/BY6SDNxTGWAqkHiz8

Replace with Scientific Term

- ✧ Document that describes how the software and associated hardware will be installed in the customer’s environment.
- ✧ Document used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.
- ✧ Actions introduced to reduce the risks of project failure if there are serious problems with the development work that are likely to lead to significant delays.
- ✧ Points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- ✧ Work products that are delivered to the customer, e.g. a requirements document for the system.

List

- Factors affecting software pricing.
- Reasons for overpricing and underpricing.
- Project plan elements.
- Project scheduling activities.
- Project scheduling problems.
- Agile planning stages.

Questions

- Project planning is conducted during the project proposal and startup only. Comment.
- What is the rationale behind development by following a plan?
- Elaborate on the need for plans supplementary to the project plan.
- You should include ---- in your plan so that if things go wrong, then your delivery schedule is not seriously disrupted. Complete.
- Give two reasons for inevitable change to plans.
- What is the course of actions required if there are serious problems with the development work that are likely to lead to significant delays?

STUDY SMART