Chapter 1- Introduction to Software Engineering

جامعة القادسية/ كلية علوم الحاسوب

وتكنولوجيا المعلومات/ قسم نظم المعلومات

الحاسوبية/ المرحلة الثالثة

❑ Introduction to Software Engineering:

- Software engineering is concerned with theories, methods and tools for professional software development.
- Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with cost-effective software development.

❑ Failure to use software engineering methods:

- It is fairly easy to write computer programs without using software engineering methods and techniques. Many companies have drifted into software development as their products and services have evolved. They do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.

❑ Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

❑ The Attributes of Good Software:

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Usability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use. (Easy to Use).

❑ Software Products:

- **Generic products**

Stand-alone systems that are marketed and sold to any customer who wishes to buy them. The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.

- **Customized/Bespoke products**

Software that is commissioned by a specific customer to meet their own needs. The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

❑Important Points:

- There are many different types of system and each requires appropriate software engineering tools and techniques for their development.
- The fundamental ideas of software engineering are applicable to all types of software system.
- Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.
- Professional societies publish codes of conduct which set out the standards of behaviour expected of their members.

Chapter 2- Software Process

جامعة القادسية/ كلية علوم الحاسوب
وتكنولوجيا المعلومات/ قسم نظم المعلومات
الحاسوبية/ المرحلة الثالثة

❑ The software process:

- A structured set of activities required to develop a software system.
- Many different software processes but all involve:
  - ➢ Specification – defining what the system should do;
  - ➢ Design and implementation – defining the organization of the system and implementing the system;
  - ➢ Validation – checking that it does what the customer wants;
  - ➢ Evolution – changing the system in response to changing customer needs.
- A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.

❑ Software process descriptions:

- When we describe and discuss processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc. and the ordering of these activities.
- Process descriptions may also include:
  - ➢ Products, which are the outcomes of a process activity.
  - ➢ Roles, which reflect the responsibilities of the people involved in the process

❑ Software Process Models:

- **The Waterfall Model**

➢ Separate and distinct phases of specification and development.

- **Prototype Development**

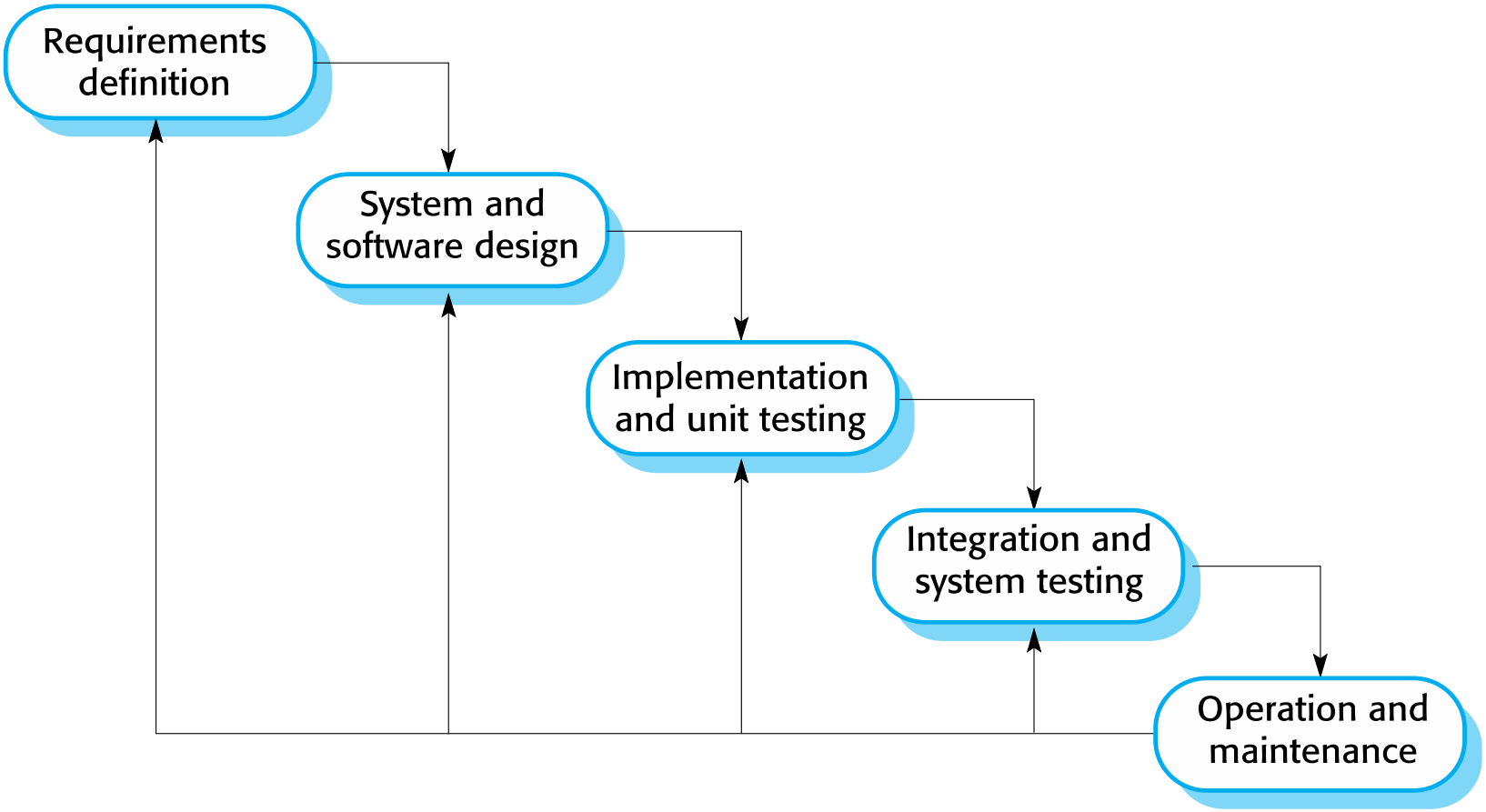
➢ Specification, development and validation are interleaved.

- **Component-Based Development**

➢ The system is assembled from existing configurable components.

- In practice, most large systems are developed using a process that incorporates elements from all of these models.

❑ The Draw of Waterfall Model:



❑ Important Points:

- Software processes are the activities involved in producing a software system. Software process models are abstract representations of these processes.
- General process models describe the organization of software processes.
  - ➢ Examples of these general models include the ‘waterfall’ model, prototype development, and reuse-oriented development.
- Requirements engineering is the process of developing a software specification.

❑Important Points:

- Design and implementation processes are concerned with transforming a requirements specification into an executable software system.
- Software validation is the process of checking that the system conforms to its specification and that it meets the real needs of the users of the system.
- Software evolution takes place when you change existing software systems to meet new requirements. The software must evolve to remain useful.
- Processes should include activities such as prototyping and incremental delivery to cope with change.
- Processes may be structured for iterative development and delivery so that changes may be made without disrupting the system as a whole.

Chapter 3- Introduction to *Project Management*

جامعة القادسية/ كلية علوم الحاسوب وتكنولوجيا المعلومات/ قسم نظم المعلومات الحاسوبية/ المرحلة الثالثة

Introduction

- **What is Management?**
- Basically, the management involves the following activities:
- – **Planning**– deciding what is to be done
- – **Organizing**– making arrangements
- – **Staffing**– selecting the right people for the job
- – **Directing**– giving instructions
- – **Monitoring**– checking on progress
- – **Controlling**– taking action to remedy hold-ups
- – **Innovating**– coming up with new solutions
- – **Representing**– liaising with users, etc.

Introduction

– **What is Project Management?**

- Project Management is the art of maximizing the probability that a project delivers its goals on **Time**, to **Cost/Budget** and at the required **Quality**.
- The art of planning for the future has always been a human trait. In essence a project can be captured on paper with a few simple elements: a start date, an end date, the tasks that have to be carried out and when they should be finished, and some idea of the resources (people, machines etc) that will be needed during the course of the project.

- Project management is the **application of knowledge, skills, tools, and techniques** to project activities to meet project requirements. Project management is accomplished through the use of the processes such as: initiating, planning, executing, controlling, and closing. It is important to note that many of the processes within project management are iterative in nature. This is in part due to the existence of and the necessity for progressive elaboration in a project throughout the project life cycle; i.e., the more you know about your project, the better you are able to manage it.
- The term project management is sometimes used to describe an organizational approach to the management of ongoing operations. This approach, more properly called **management by projects**, treats many aspects of ongoing operations as projects to apply project management techniques to them.

- **Almost any human activity that involves carrying out a non-repetitive task can be a project.** So we are all project managers! We all practice project management (PM). But there is a big difference between carrying out a very simple project involving one or two people and one involving a complex mix of people, organizations and tasks.

– **What is Software Project Management?**

- When the plan starts to involve different things happening at different times, some of which are dependent on each other, plus resources required at different times and in different quantities and perhaps working at different rates, the paper plan could start to cover a vast area and be unreadable. (adjust plan)

- Nevertheless, the idea that complex plans could be analyzed by a computer to allow someone to control a project is the basis of much of the development in technology that now allows projects of any size and complexity, not only to be planned, but also modeled to answer 'what if?' questions.
- The original programs and computers tended to produce answers long after an event had taken place. Now, there are many project planning and scheduling programs that can provide real time information, as well as linking to risk analysis, time recording, and costing, estimating and other aspects of project control.
- But **computer programs are not project management**: **they are tools for project managers to use**. Project management is all that mix of components of control, leadership, teamwork, resource management etc that goes into a successful project.

- Project managers can be found in all industries. Their numbers have grown rapidly as industry and commerce has realized that much of what it does is project work and as project-based organizations have started to emerge, project management is becoming established as both a professional career path and a way of controlling business.
- So opportunities in project management now exist not only in being a project manager, but also as part of the support team in a project or program office or as a team leader for part of a project. There are also qualifications that can be attained through the professional associations.

– What is a Project?

- A project is **an activity with specific goals which takes place over a finite period** of time.
- "A temporary organization that is needed to produce a unique and pre-defined outcome or result at a pre-specified time using pre-determined resources".
- Projects are often implemented as a means of achieving an organization's strategic plan. Operations and projects differ primarily in that operations are ongoing and repetitive while projects are temporary and unique.
- A project can thus be defined in terms of its distinctive characteristics—***a project is a temporary endeavor undertaken to create a unique product or service.***

- ***Temporary*** means that every project has a definite beginning and a definite end.
- ***Unique*** means that the product or service is different in some distinguishing way from all other products or services.
- For many organizations, projects are a means to respond to those requests that cannot be addressed within the organization's normal operational limits.
- Projects are undertaken at all levels of the organization. They may involve a single person or many thousands.
- Their duration ranges from a few weeks to more than five years. Projects may involve a single unit of one organization or may cross organizational boundaries, as in joint ventures and partnering.

The Triangle–Triple Constraint of Project Management

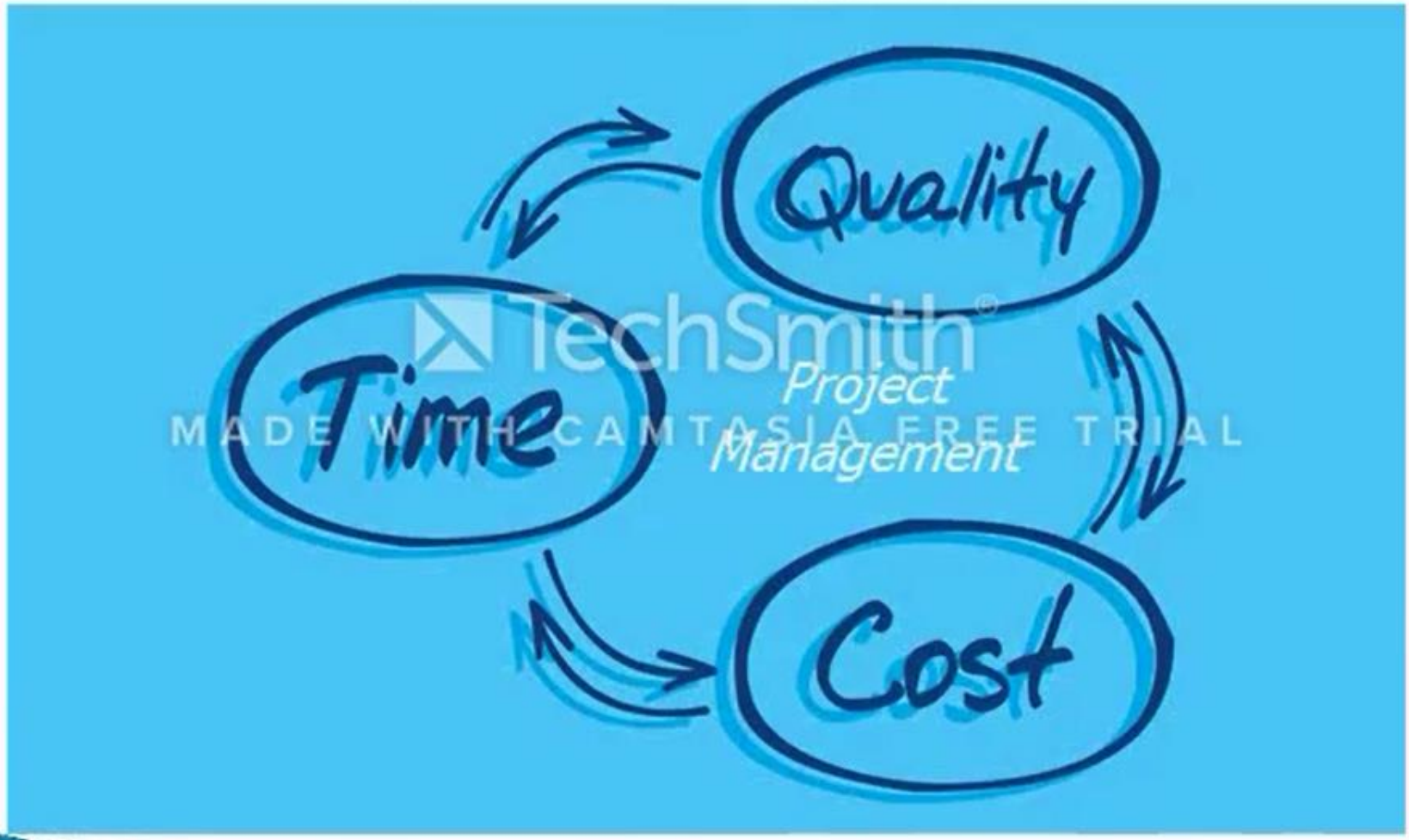


Figure 1

The Triangle–Triple Constraint of Project Management

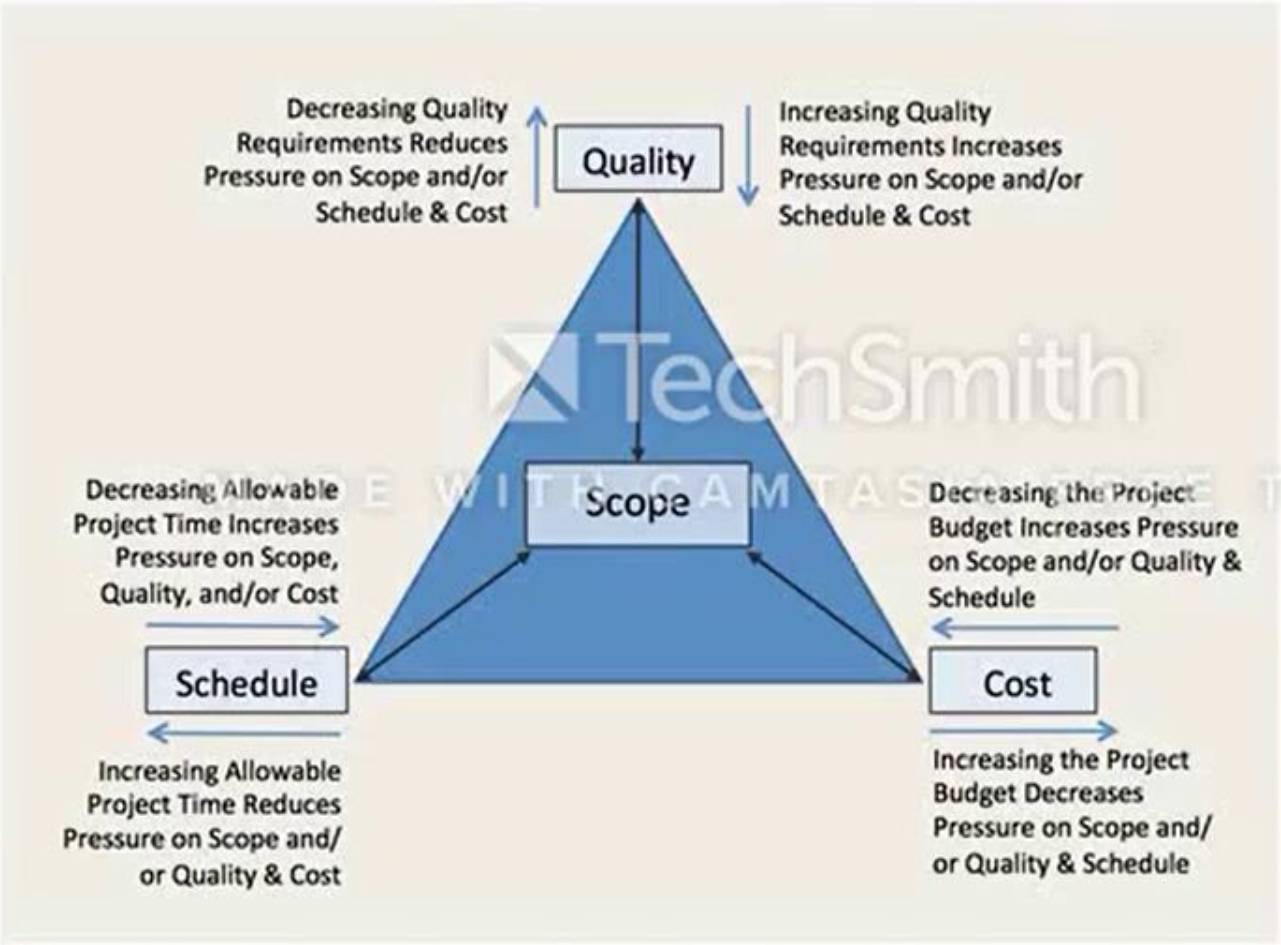
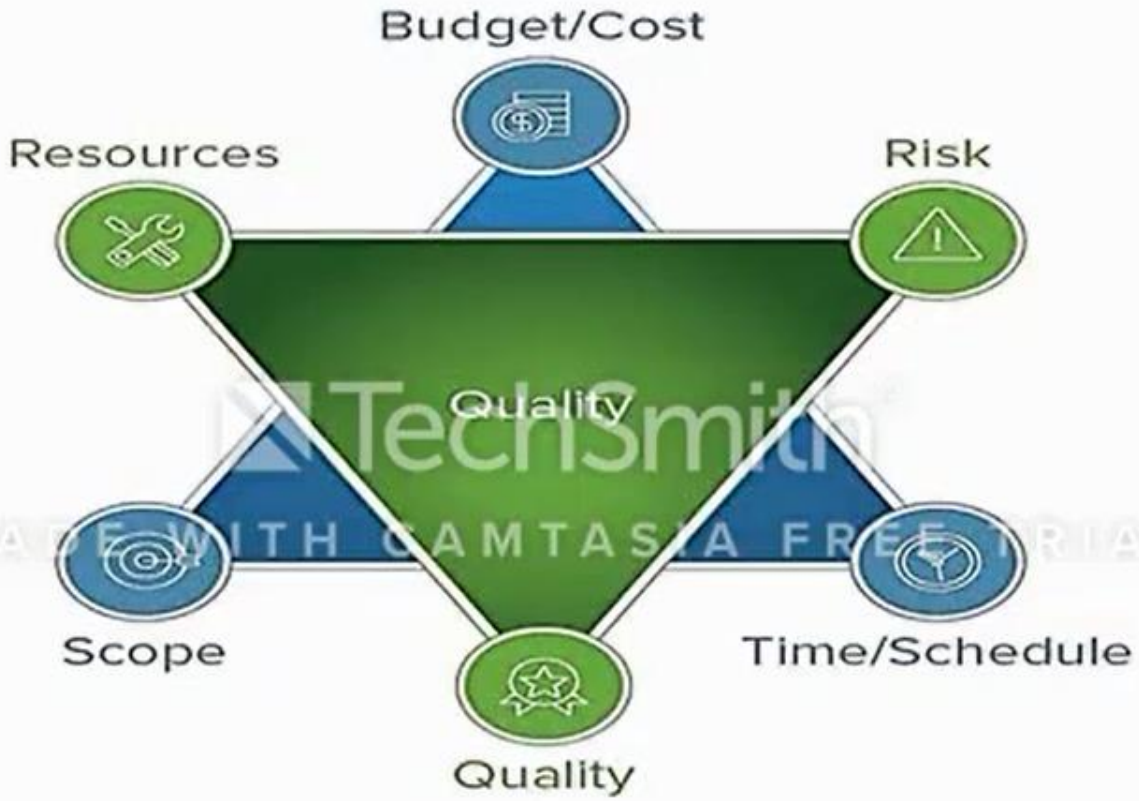


Figure 2

Triple Constraint

- Triple constraints describes the basic elements of a project. There are also some elements that play significant roles. These are Quality Customer or Sponsor satisfaction. Some people believe that quality considerations, including customer satisfaction must be inherent in setting the scope, time and cost goals of a project. Thus , a good project management meets not only the triple constraint (scope, time and cost) but also the quality and customer satisfaction.

Double Triangle Constraint



The "double triangle" diagram of project constraint sometimes measures "client satisfaction" as well as "team satisfaction," as two of the three other prongs on the new triangle, along with "processes." This notation can be especially helpful when delivering a software solutions for a client.

Figure 3

Project Management Framework

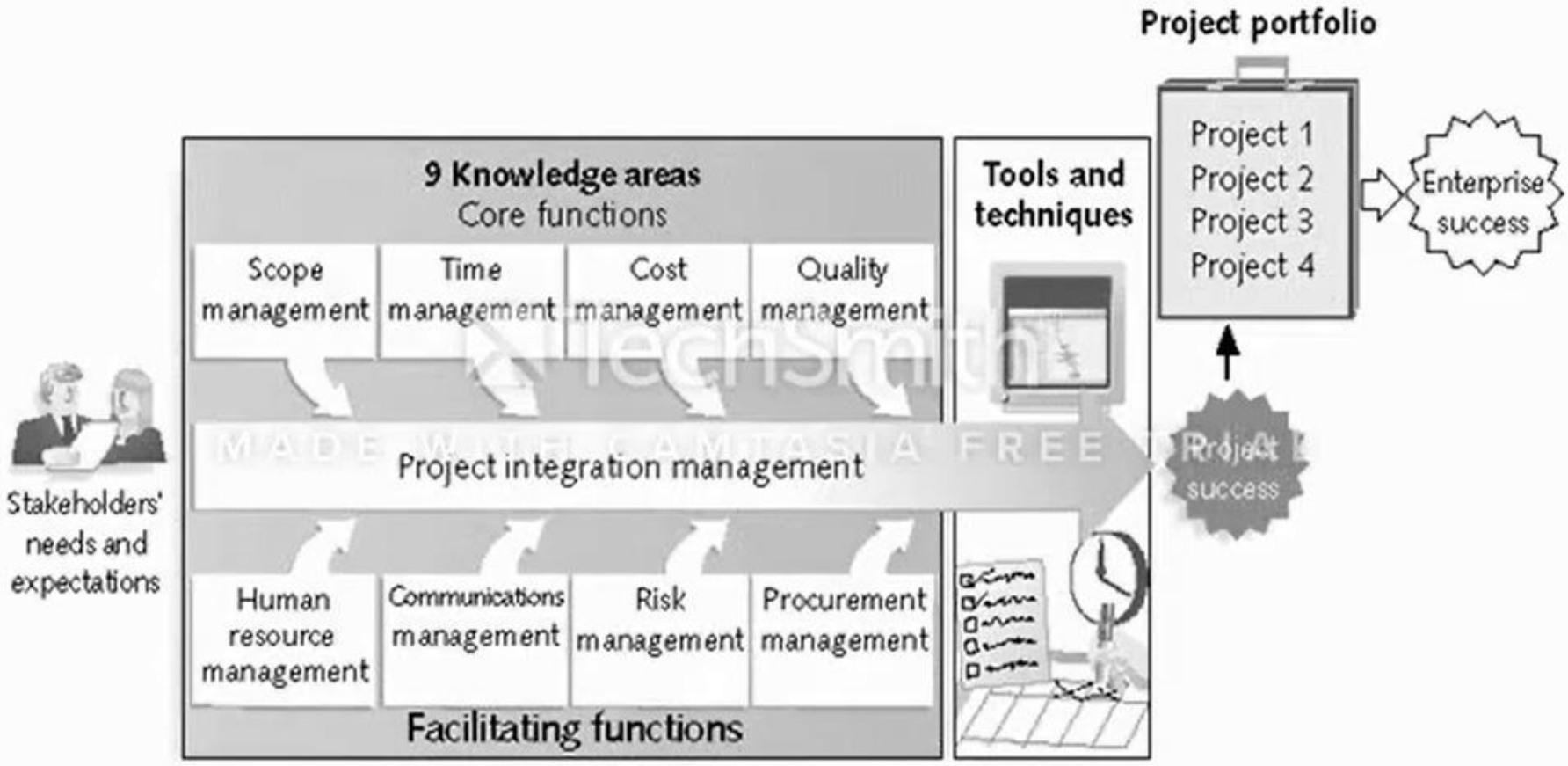


Figure 4

Key elements of the Frame work

- Project stakeholders.
- Project management knowledge areas.
- Project management tools and techniques.

Chapter 4- Risk Management

جامعة القادسية/ كلية علوم الحاسوب وتكنولوجيا المعلومات

/ قسم نظم المعلومات الحاسوبية/ المرحلة الثالثة

- Risk management is one of the main activities of project managers. It involves anticipating risks that might affect the project schedule or the quality of the software being developed and taking action to avoid these risks. The results of the risk analysis should be documented in the project plan along with an analysis of the consequences of a risk occurring. There are three categories of risk:
  - Project risks. These risks affect the entire project schedule.
  - Product risks. Product risks have influence on the quality and performance of the software.
  - Business risks. These risks affect the organization developing or procuring the software.

- Project manager has to get ready for risks, understand the impact of these risks on the project, the product and the business, and take steps to avoid these risks. Project manager should prepare contingency plans so that, if the risks occur, he can take immediate actions.
- The risk management process is shown in Figure-1 It has four stages:
  - Risk identification. Possible project, product and business risks are identified.
  - Risk analysis. Using risk analysis the probability and consequences of risks are analyzed.
  - Risk planning. The objective of risk planning is to avoid the risks or minimize its effects.
  - Risk monitoring. The identified risk is continuously assessed and the avoidance strategies are revised as more information becomes available about the risk.

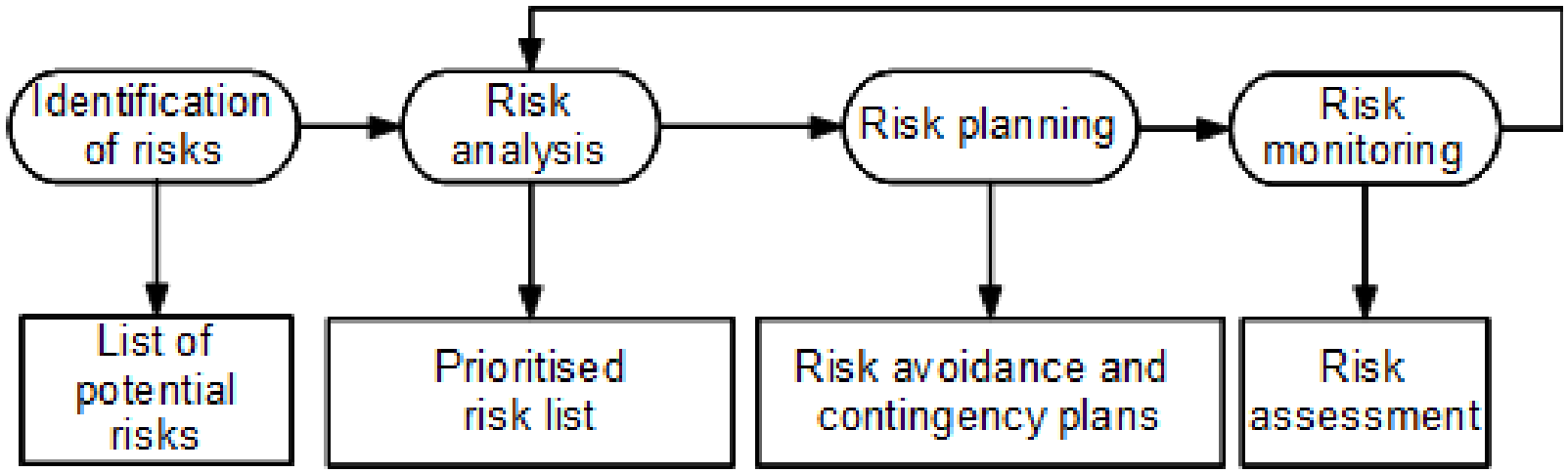


Fig-(1)

- The risk management process is an iterative process which continues throughout the project. As more information about the risks becomes available, the risks have to be reanalyzed and the avoidance and contingency plans has to be also modified. The outcomes of the risk management process should be documented in a risk management plan.

❑ Risk identification:

- The first stage of risk management is the identification of risks. It is concerned with discovering possible risks to the project. Risk identification may be carried out as a team work based on the experience of team members. Helping the process risks are usually classified into separate groups:
  - ➢ Technology risks. These risks are related to the software or hardware technologies that are used to develop the system.
  - ➢ People risks. Risk that are associated with the people in the development team.
  - ➢ Organizational risks. Risk that derive from the changes in organizational environment.
  - ➢ Tools risks. Risk that derive from the CASE tools and other support software used in the development process.
  - ➢ Requirements risks. Risk that derive from changes to the customer requirements and the process of managing the requirements change.
  - ➢ Estimation risks. Risk that derive from the management estimates of the system characteristics and the resources required to build the system.

❑ Risk analysis:

- During the risk analysis process, the probability and the effects of identified risks are determined. This process is mainly based on the experience of project managers. The risk estimates are generally assigned to different bands:
  - ➢ The probability of the risk might be assessed as very low (<10%), low (10–25%), moderate (25-50%), high (50–75%) or very high (>75%).

- The effects of the risk might be assessed as catastrophic, serious, tolerable or insignificant.

- Once the risks have been analyzed and ranked, it is necessary to decide which risks are most significant. In general, the catastrophic risks and the serious risks having more than a moderate probability of occurrence should always be considered.

❑ Risk planning:

- The risk planning process identifies strategies to manage the identified risks. There is no general procedure to establish risk management plans. It is mostly based on the judgment and experience of the project manager.
- The risk management strategies have three categories:
  - **Avoidance strategies:** Following these strategies means that the probability that the risk will arise will be reduced.
  - **Minimization strategies:** The objective of these strategies is to reduce the effect of risk. An example of a risk minimization strategy is that for staff illness.
  - **Contingency plans:** These are the strategies used when risk arises.

❑ Risk monitoring:

- The purpose of risk monitoring risk is to constantly assess each of the identified risks to decide whether its probability and effects have changed. Usually this cannot be observed directly and other factors should be monitored to detect the changes.

- Table-(1) give some examples of factors that may be helpful in assessing different types of risks.

Risk type	Potential indicators
Technology	One of the hardware supplier companies becomes bankrupt
People	Software engineer, who gets a flu in every year complains about weakness
Organizational	Poor managerial activity is seen
Tools	Using a CASE tool the first error occurs
Requirements	Customer complains about the performance of current version of software
Estimation	The status of the approved project schedule is very poor

Table-(1) Risk Factors

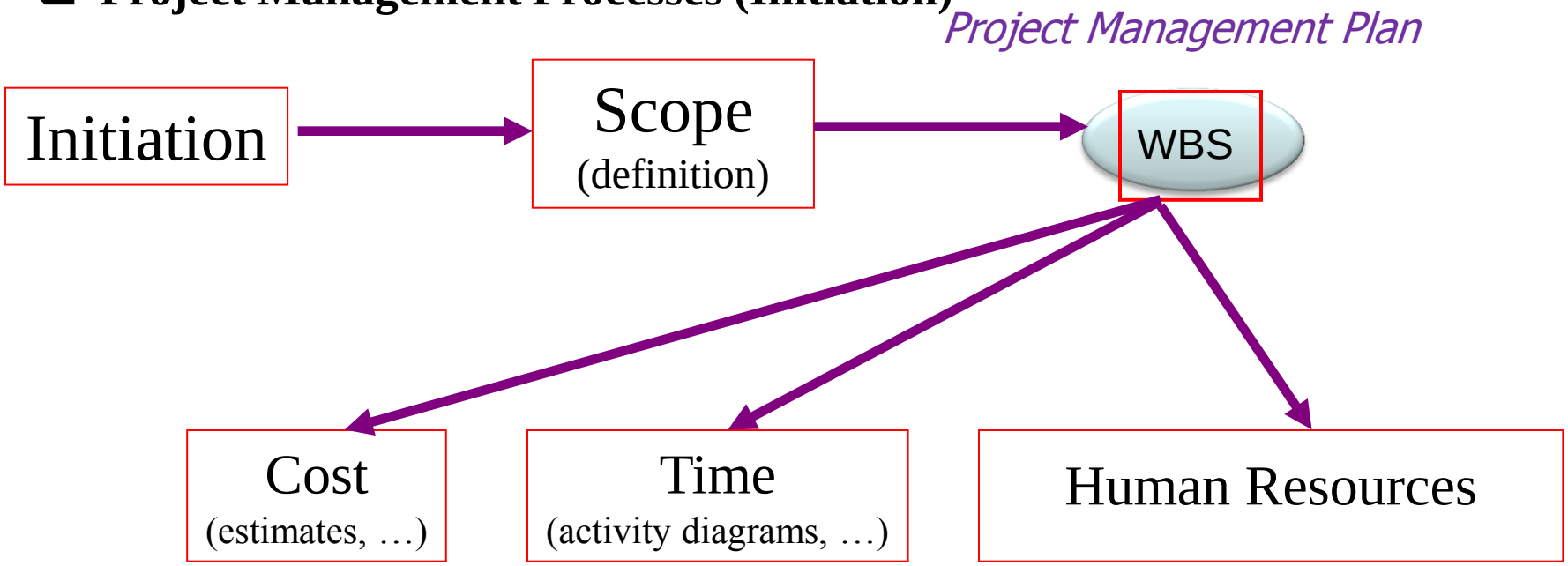
Chapter 5- Project Planning and Scheduling

جامعة القادسية/ كلية علوم الحاسوب وتكنولوجيا المعلومات

/ قسم نظم المعلومات الحاسوبية/ المرحلة الثالثة

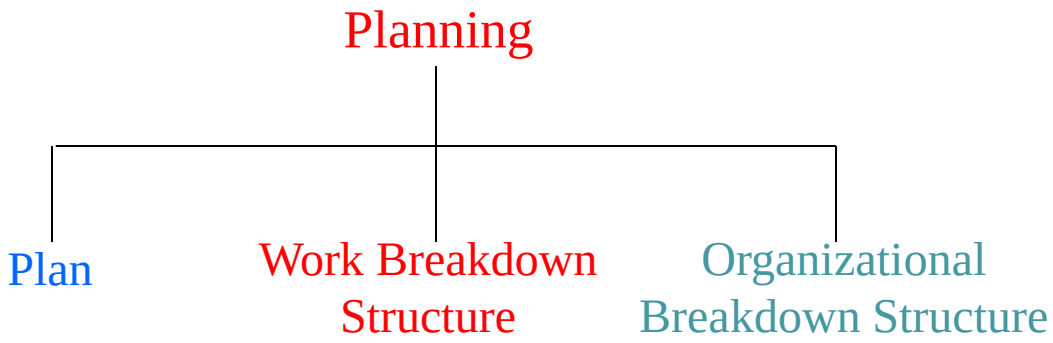
Project Planning and Scheduling

- **Project Management Processes (Initiation)**



- **Elements of Project Managements:**

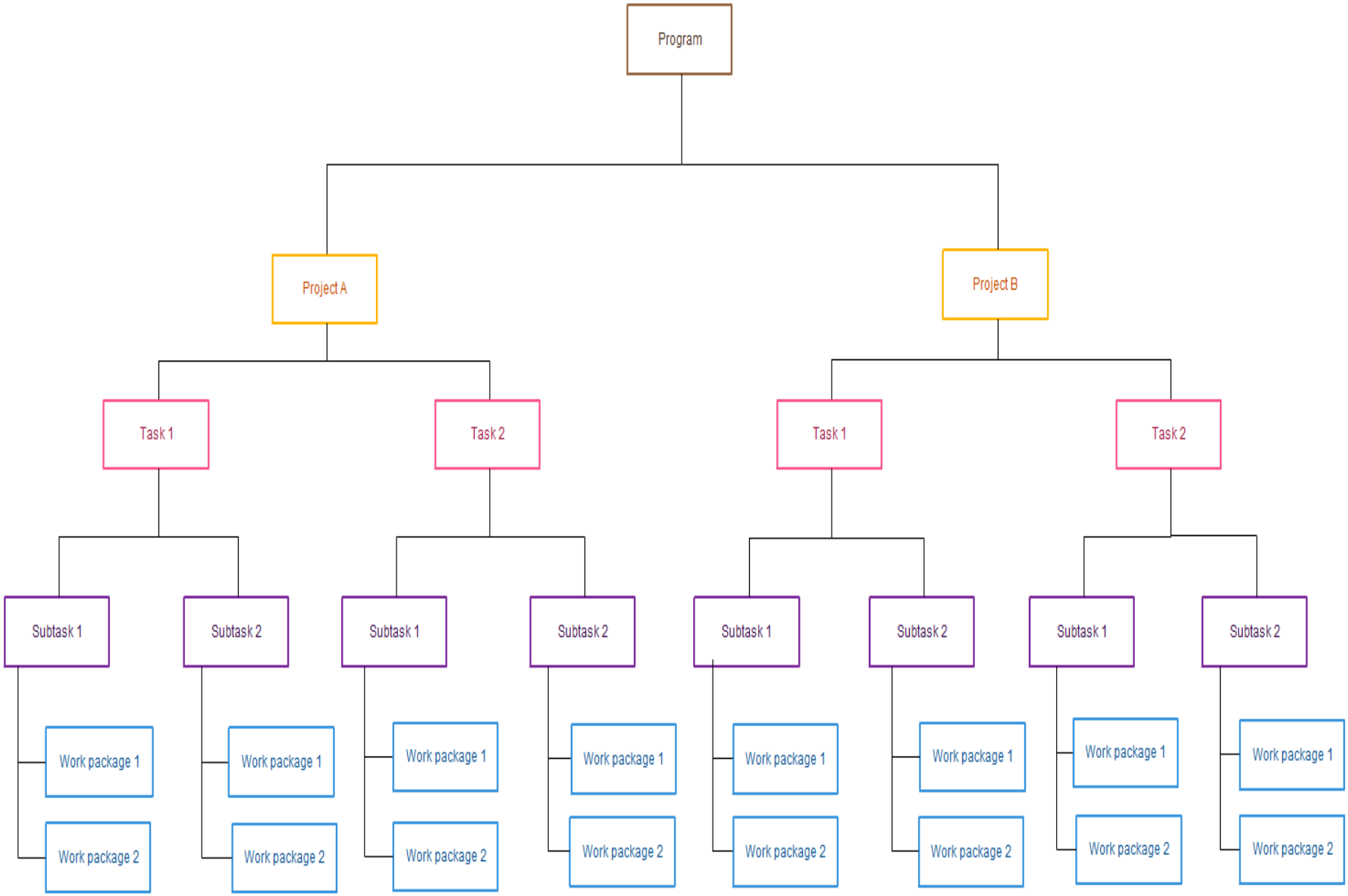
1- Planning



Elements of Project Management
Work Breakdown Structure (WBS)

- **WBS breaks down project** into major components (modules). In general WBS divide complex projects into smaller and manageable tasks.
- Modules are further broken down into subcomponents, components, activities, and finally, into individual tasks. These components can be easily supervised and estimated and these components are called **work package.**
- Identifies activities, tasks, resource requirements and relationships between modules and activities.
- Helps avoid duplication of effort.
- Basis for project development, management , schedule resources and modifications.

Work Breakdown Structure



WBS Decomposition

WBS LEVEL 0:

1. Bicycle 100

WBS LEVEL 1:

1. Bicycle	
1.1 Frame Set	15
1.2 Crank Set	5
1.3 Wheels	30
1.4 Braking System	5
1.5 Shifting System	5
1.6 Integration	35
1.7 Project Mgt.	5

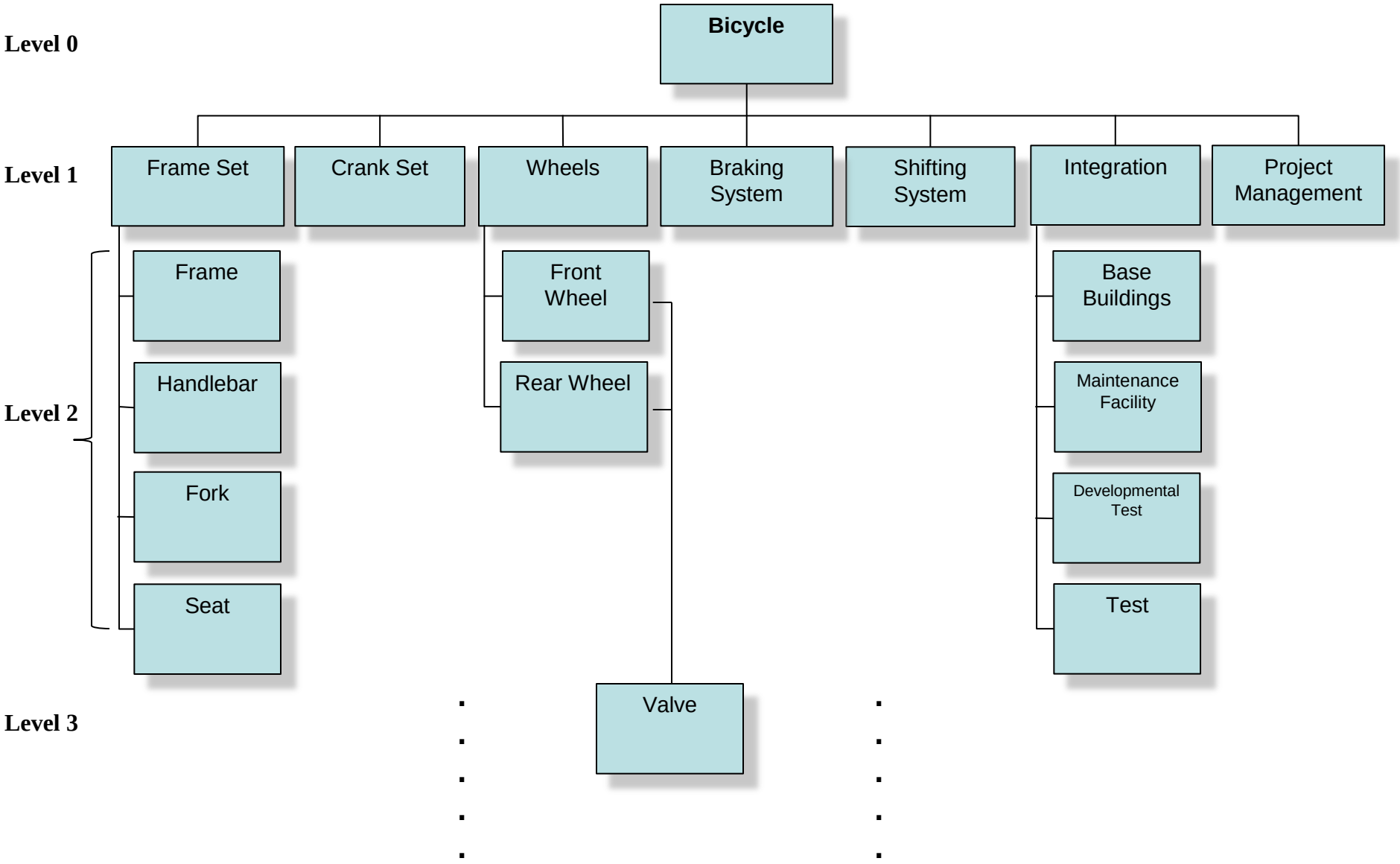
	100

WBS LEVEL 2:

1. Bicycle	
1.1 Frame Set	
1.1.1 Frame	7
1.1.2 Handlebar	2
1.1.3 Fork	3
1.1.4 Seat	3
1.2 Crank Set	5
1.3 Wheels	
1.3.1 Front Wheel	13
1.3.2 Rear Wheel	17
1.4 Braking System	5
1.5 Shifting System	5
1.6 Integration	
1.6.1 Concept	3
1.6.2 Design	5
1.6.3 Assembly	10
1.6.4 Testing	17
1.7 Project Mgt.	5

	100

Elements of Project Management
Work Breakdown Structure



Elements of Project Management
Project Scheduling

- Project schedule evolves from planning documents, with focus on timely completion.
- Critical element in project management – source of most conflicts and problems.
- Decide end date & start date of each activity.
- Sequential relationship between each activities.
- Allocation of resources.
- Tools used in project scheduling are: Gantt chart and (Critical Path Method (CPM)/Program Evaluation Review Technique (PERT)) can be useful.
- Schedule development steps:

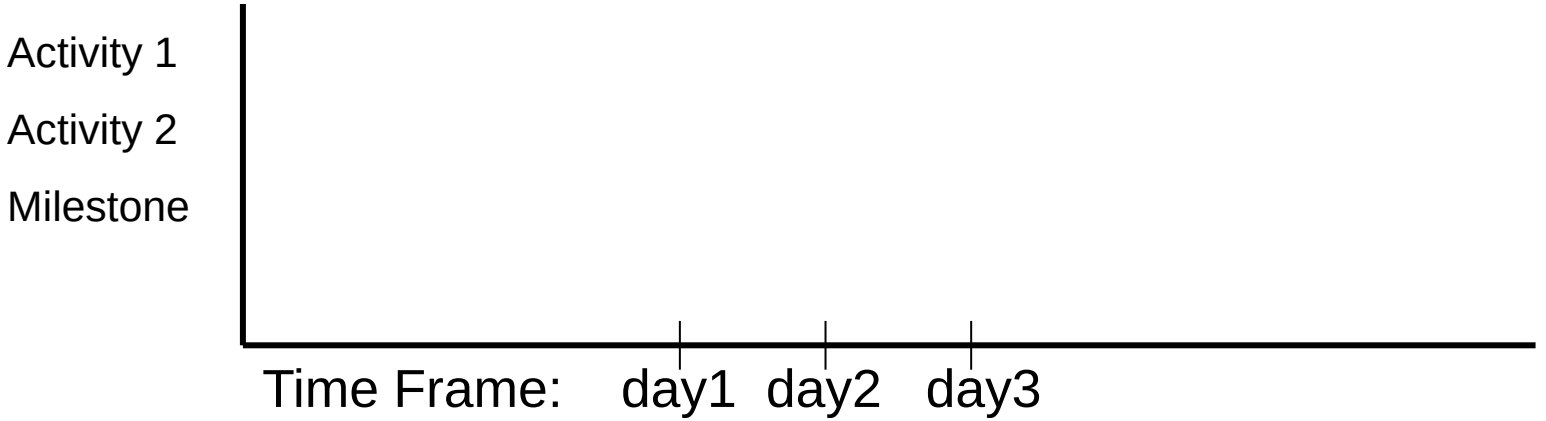
  1. Define activities,
  2. Sequence activities,
  3. Estimate activity times,
  4. Develop schedule.

Elements of Project Management
Gantt Chart

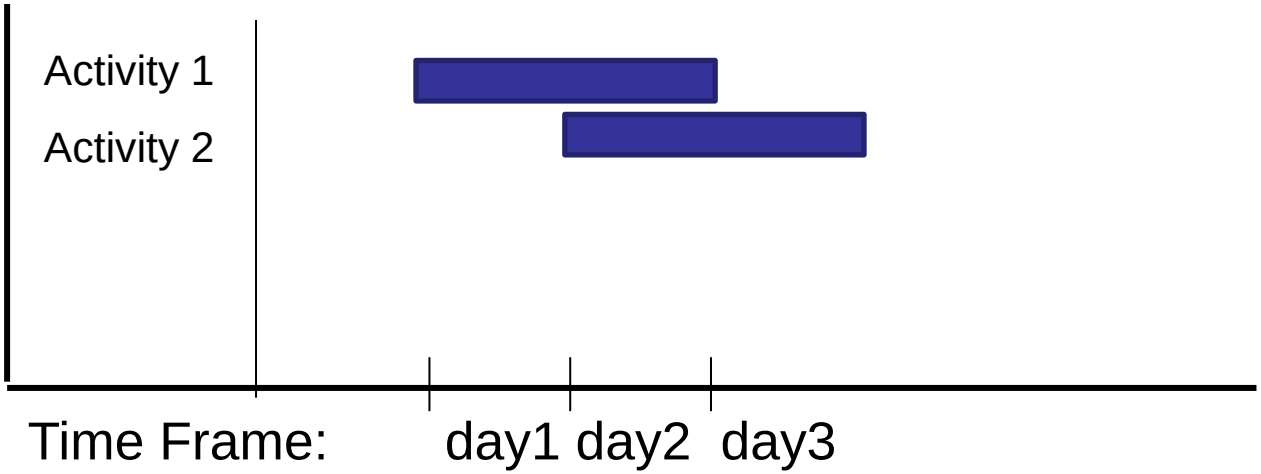
- Popular, traditional technique, also known as a ***bar chart*** -developed by Henry Gantt.
- Direct precursor of CPM/PERT for monitoring work progress.
- A ***visual display of project schedule*** showing activity start and finish times and where extra time is available.
- Suitable for projects with few activities and precedence relationships.
- Visual scheduling tool
- Graphical representation of information
- Show dependencies between tasks, personnel, and other resources allocations
- Track progress towards completion

Building a Gantt Chart

- List all tasks and milestones from the project along the vertical axis
- List time frame along the horizontal axis

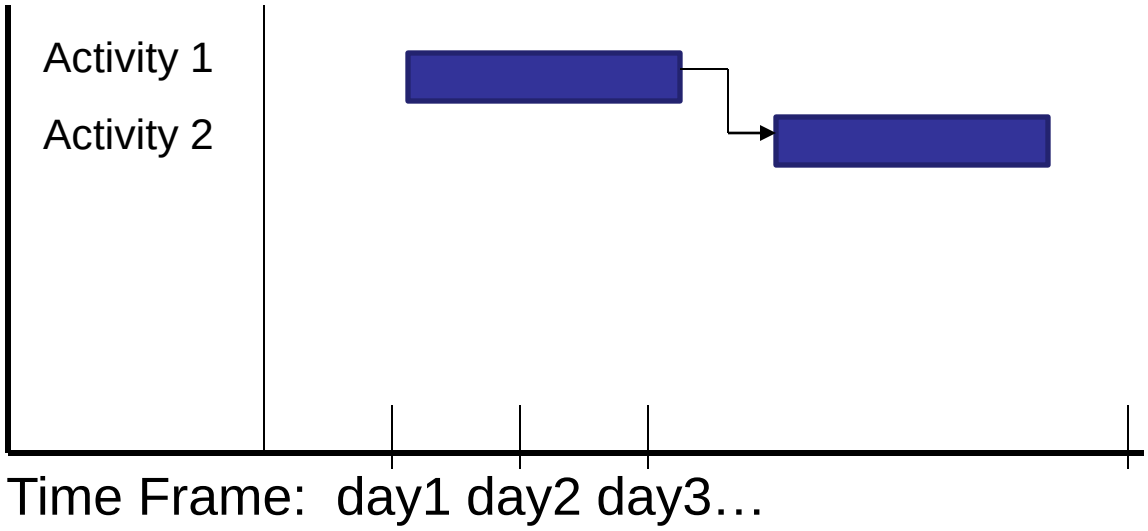


- Activities: Create box the length of each activity time duration
  E.g., activity one is scheduled from day1-day3.



Building a Gantt Chart

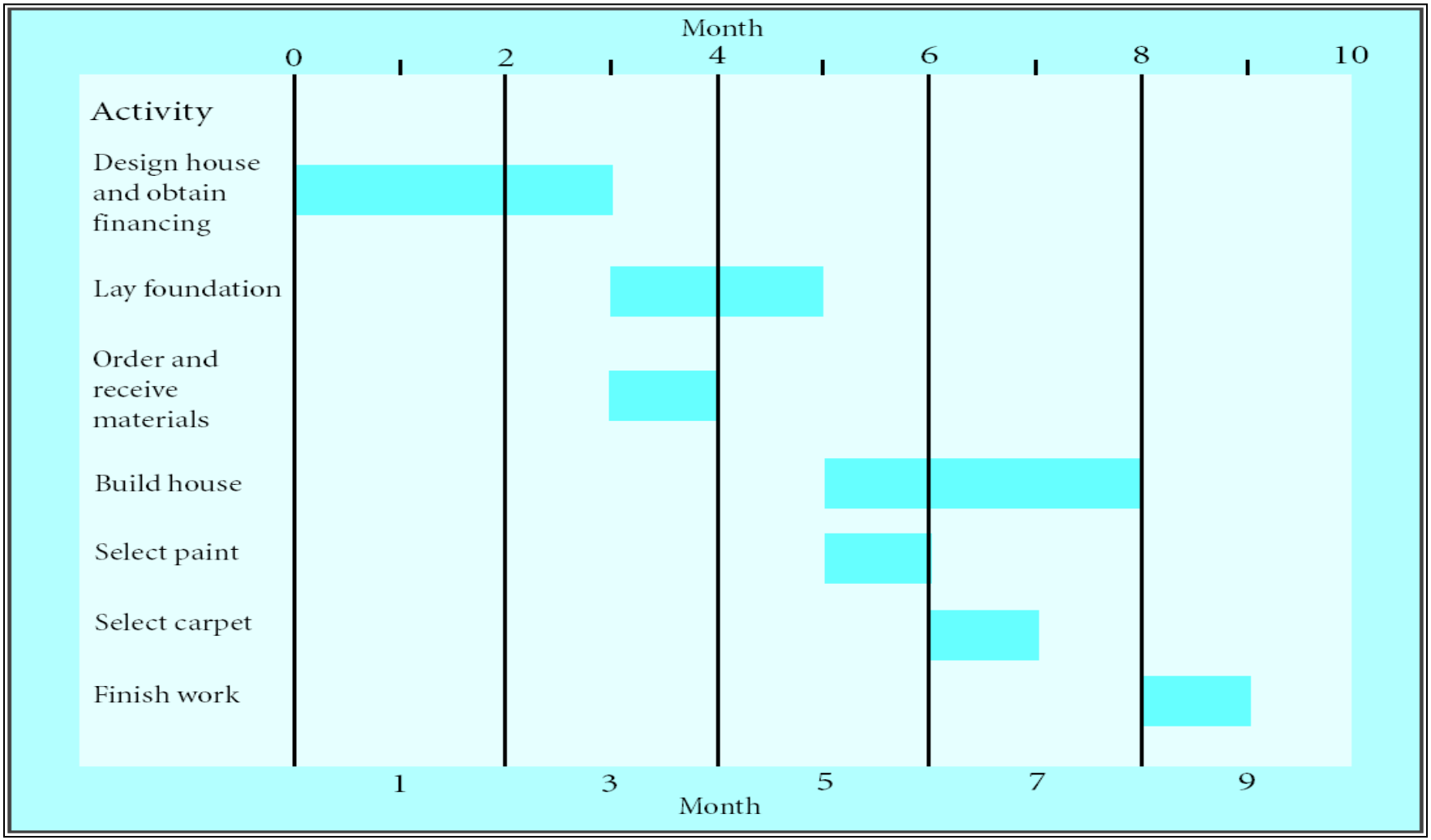
- Dependencies: Show dependencies between activities with arrows
  E.g., activity 2 cannot start until activity 1 is complete



Sequence of Activities of The Project-House Building

Number	Activity	Predecessor	Duration
1	Design house and obtain financing	--	3 months
2	Lay foundation	1	2 months
3	Order and receive materials	1	1 month
4	Build house	2,3	3 months
5	Select paint	2, 3	1 month
6	Select carper	5	1 month
7	Finish work	4, 6	1 month

Gantt Chart for House Building Project



Example on Gantt Chart

Example1: The table below explain the activities and duration time for implement the project according to the following:

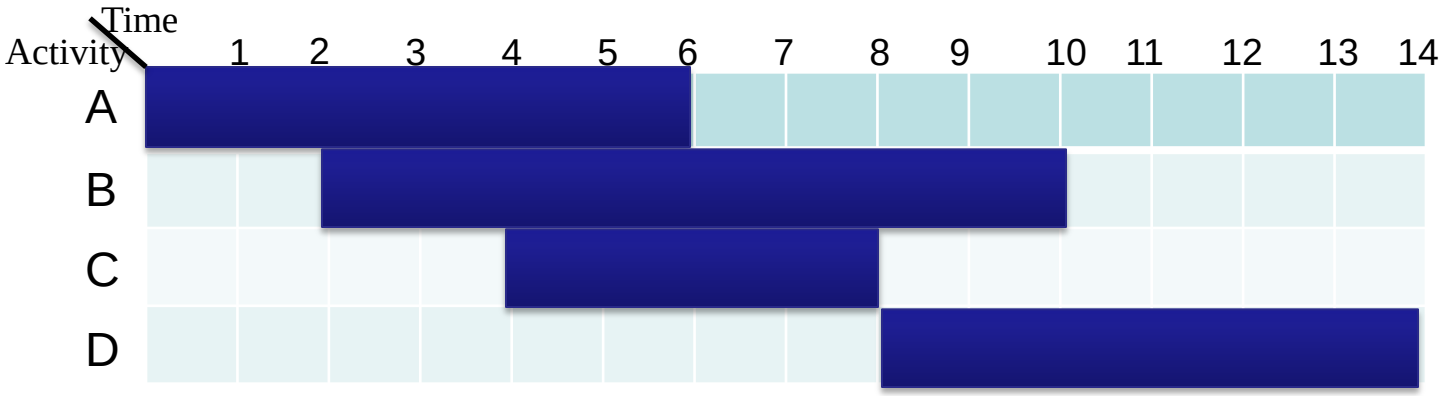
1- Activity B start after two weeks from Activity A started.

2- Activity C start after two weeks from Activity B started.

3- Activity D start after Activity C ended.

Activity	Time/Weeks
A	6
B	8
C	4
D	6

Example on Gantt Chart



The total project time is 14

Elements of Project Management

3- Controlling:

It's a process which is performed to control the project execution. In this potential problems are identified and corrective measures are taken in a timely manner.

4- Directing:

It's a function of a project leader to give instruction to subordinates, supervise their work and respond to their reports.

5- Coordinating:

It's a process of interaction between departments of a projects, ex: planning, execution, designing…

6- Organizing:

It's an integration of resources such that, desired objective an achieve.

Project Scheduling
Critical Path Method (CPM)

❑ PERT*: Program Evaluation and Review Technique*

- *Developed to handle uncertain activity times*

❑ CPM: *Critical Path Method*

- Developed for industrial projects for which activity times generally were known
- Today's project management software packages have combined the best features of both approaches.

Project Scheduling
Critical Path Method (CPM)

- **PERT** and **CPM** have been used to plan, schedule, and control a wide variety of projects:
  - Construction of buildings and highways
  - Maintenance of large and complex equipment
  - Design and installation of new systems
  - A complicating factor in carrying out the activities is that some activities depend on the completion of other activities before they can be started.

The Network Diagram

- *Network (precedence) diagram* – diagram of project activities that shows sequential relationships by the use of arrows and nodes.
- *Activities* – steps in the project that consume resources and/or time.
- *Events* – the starting and finishing of activities, designated by nodes.
- *Path*- Sequence of activities that leads from the starting node to the finishing node
- *Critical path*- The longest path; determines expected project duration
- *Critical activities*- Activities on the critical path
- *Slack*- Allowable slippage for path; the difference the length of path and the length of critical path

Time Estimates & Computing Algorithm

- Deterministic
  - Time estimates that are fairly certain
- Probabilistic
  - Estimates of times that allow for variation
- Network activities
  - **ES:** **E**arliest **S**tart
  - **EF:** **E**arliest **F**inish
  - **LS:** **L**atest **S**tart
  - **LF:** **L**atest **F**inish
- Used to determine
  - Expected project duration
  - Slack time
  - Critical path

Determining the Project Schedule

Perform a Critical Path Analysis

Earliest start (ES) =	***earliest time at which an activity can start, assuming all predecessors have been completed***
Earliest finish (EF) =	***earliest time at which an activity can be finished***
Latest start (LS) =	***latest time at which an activity can start so as to not delay the completion time of the entire project***
Latest finish (LF) =	***latest time by which an activity has to be finished so as to not delay the completion time of the entire project***

Determining the Project Schedule

Earliest Start and Finish Times

- Step 1: Make a forward pass through the network as follows: For each activity *i* beginning at the Start node, compute:
  - Earliest Start Time = the maximum of the earliest finish times of all activities immediately preceding activity *i*. (This is 0 for an activity with no predecessors.)
  - Earliest Finish Time = (Earliest Start Time) + (Time to complete activity *i*).

  The project completion time is the maximum of the Earliest Finish Times at the Finish node.

Determining the Project Schedule

Latest Start and Finish Times

- Step 2: Make a backwards pass through the network as follows: Move sequentially backwards from the Finish node to the Start node. At a given node, *j*, consider all activities ending at node *j*. For each of these activities, *i*, compute:
  - Latest Finish Time = the minimum of the latest start times beginning at node *j*. (For node *N*, this is the project completion time.)
  - Latest Start Time = (Latest Finish Time) - (Time to complete activity *i*).

Determining the Critical Path

- Step 3: Calculate the slack time for each activity by:

  Slack = (Latest Start) - (Earliest Start), or

  = (Latest Finish) - (Earliest Finish)

Critical Path Procedure

- **Step 1.** Develop a list of the activities that make up the project.
- **Step 2.** Determine the immediate predecessor(s) for each activity in the project.
- **Step 3.** Estimate the completion time for each activity.
- **Step 4.** Draw a project network depicting the activities and immediate predecessors listed in steps 1 and 2.

Critical Path Procedure

- **Step 5.** Use the project network and the activity time estimates to ***determine the earliest start and the earliest finish time for each activity by making a forward pass through the network***. The earliest finish time for the last activity in the project identifies the total time required to complete the project.

- **Step 6.** Use the project completion time identified in step 5 (earliest finish) as the (latest finish) time for the last activity and make ***a backward pass through the network to identify the latest start and latest finish time for each activity***.

Critical Path Procedure

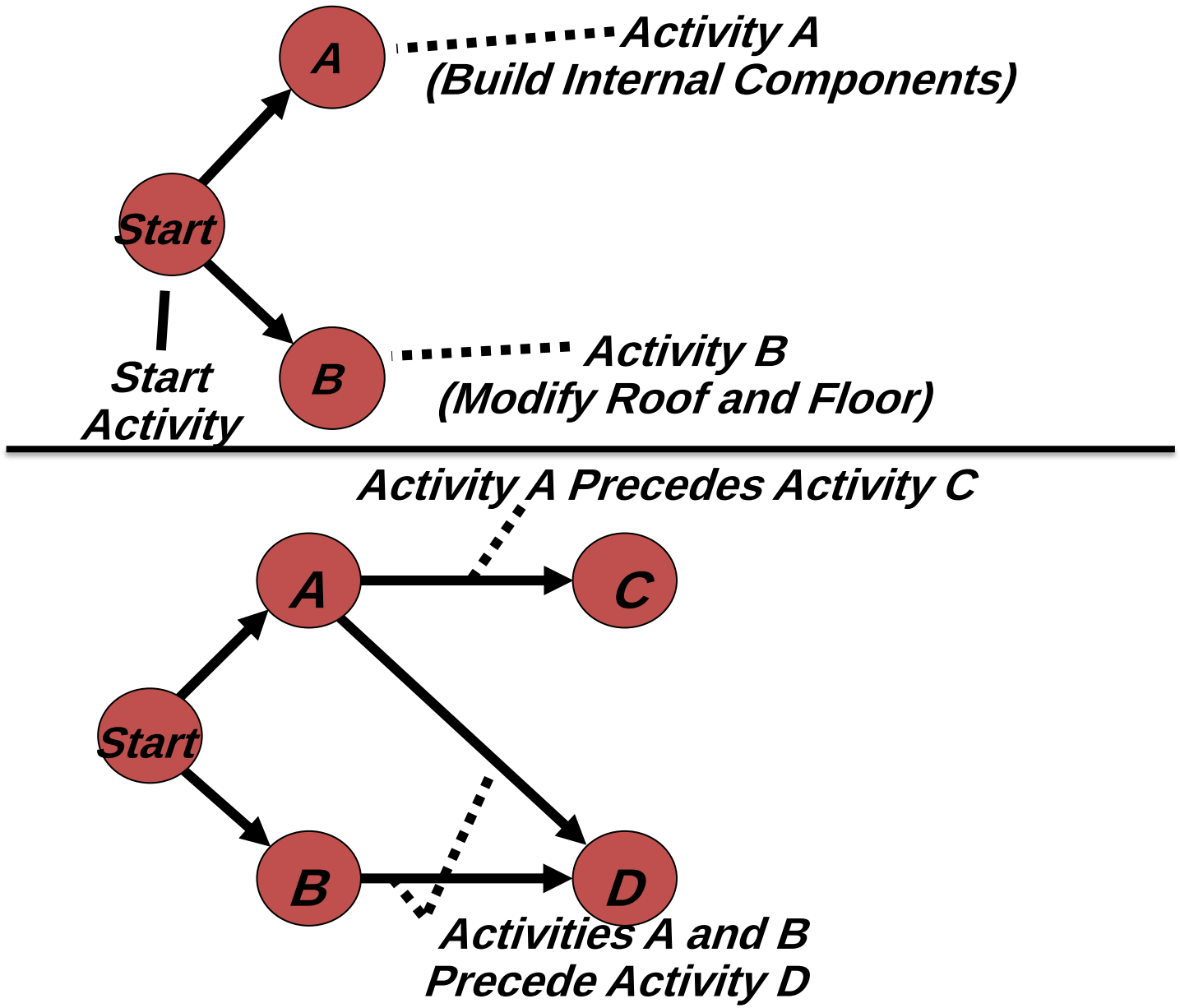
- **Step 7.** Use the difference between the latest start time and the earliest start time (**LS-ES**) for each activity to determine the slack for each activity.
- **Step 8.** Find the activities with **zero slack**; these are the critical activities.
- **Step 9.** Use the information from steps 5 and 6 to develop the activity schedule for the project.

Example(1)

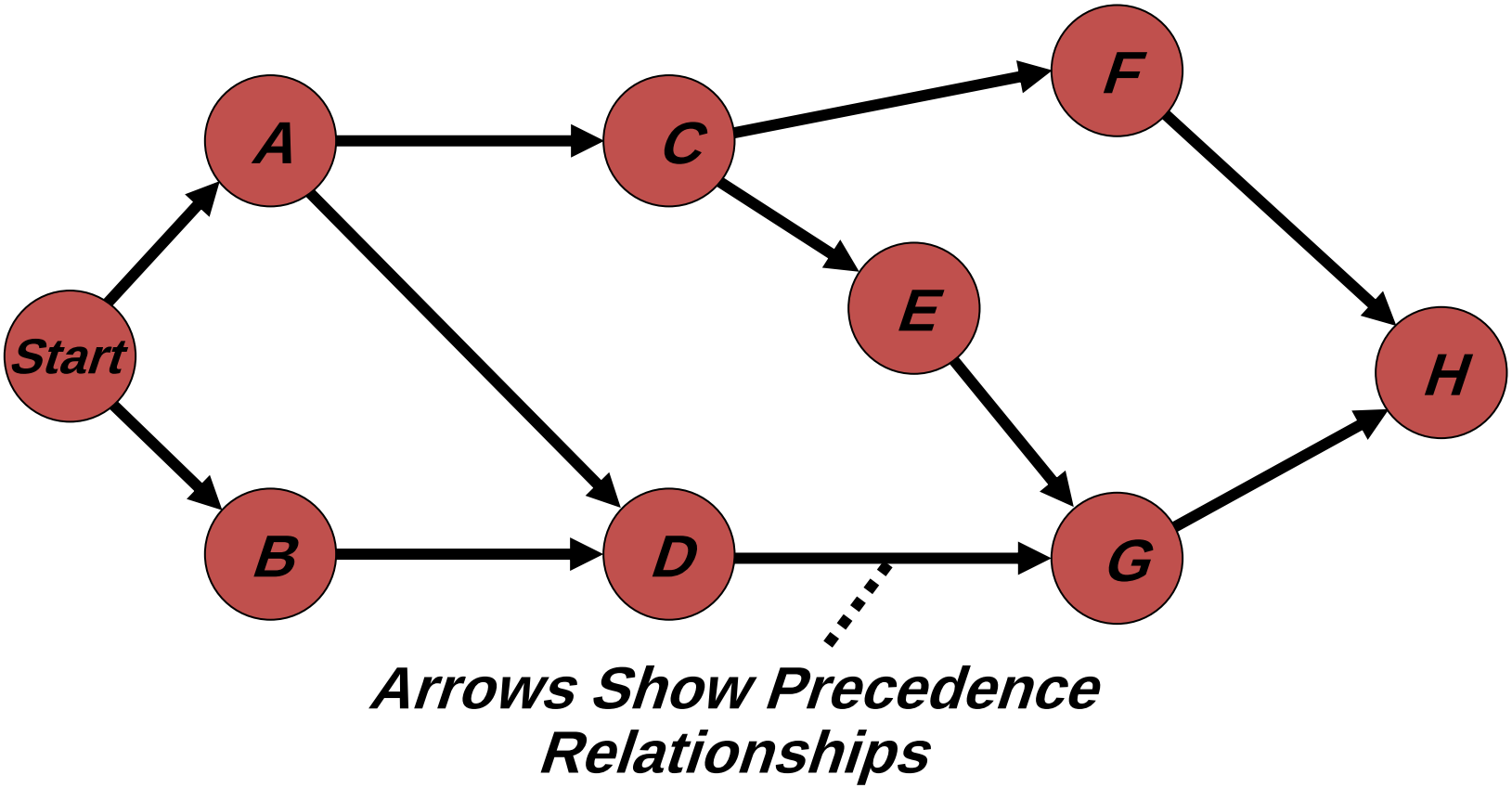
Activities and Predecessors

Activity	Description	Time (weeks)	Immediate Predecessors
A	Build internal components	2	—
B	Modify roof and floor	3	—
C	Construct collection stack	2	A
D	Pour concrete and install frame	4	A, B
E	Build high-temperature burner	4	C
F	Install pollution control system	3	C
G	Install air pollution device	5	D, E
H	Inspect and test	2	F, G
Total Time (weeks)		**25**	

Node with Arrow Network



Show Precedence Relationships on Network

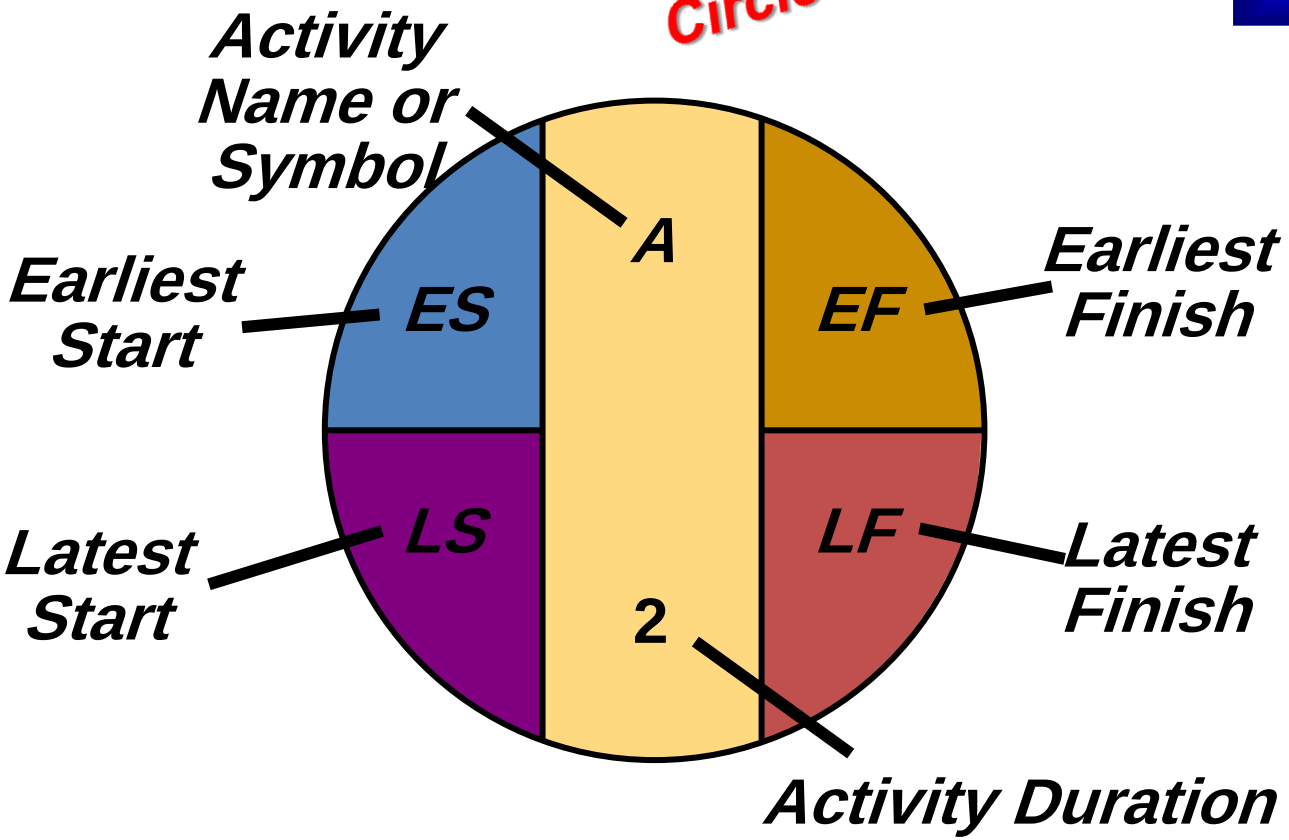


Determining the Project Schedule

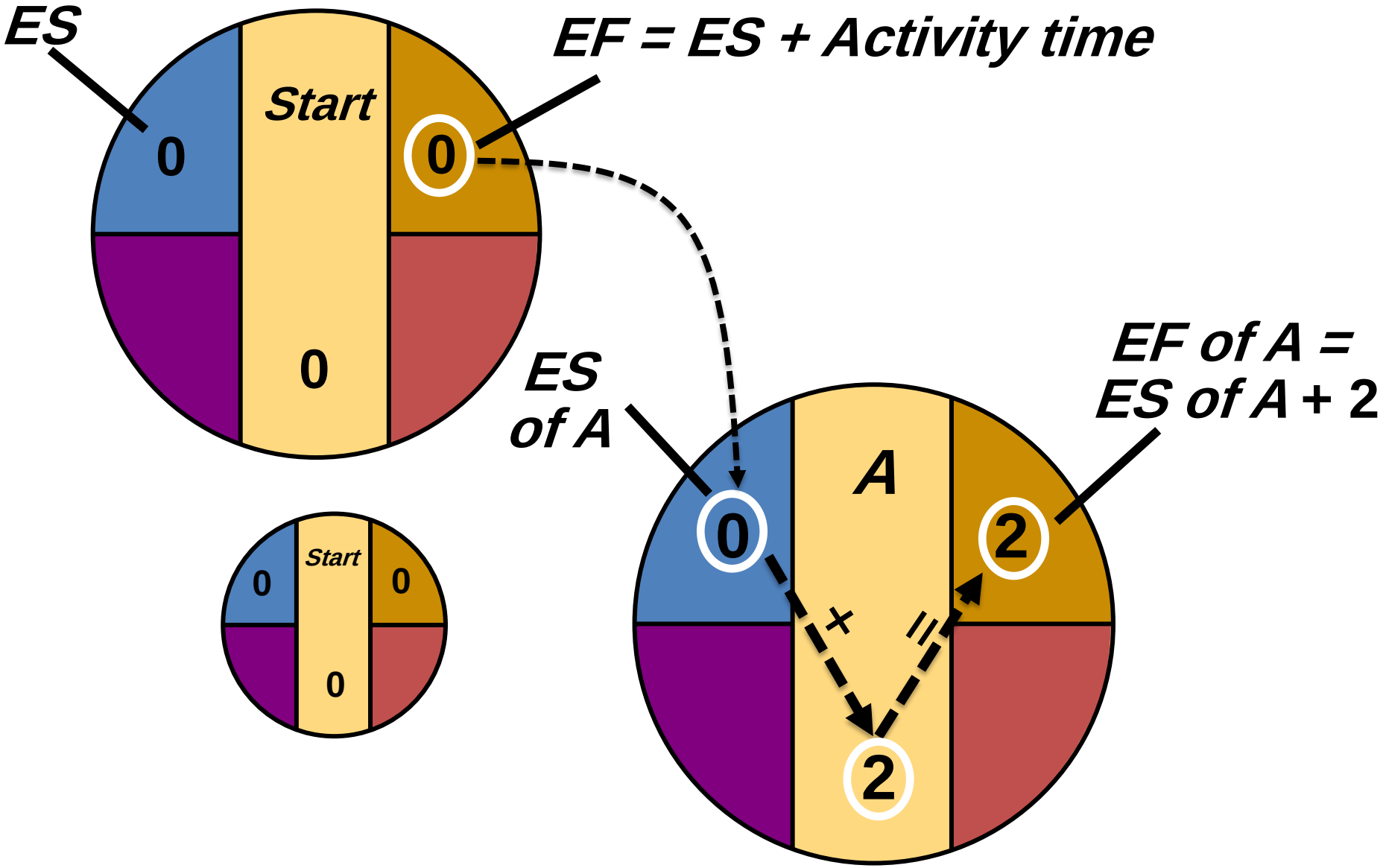
Perform a Critical Path Analysis

A	ES	EF
2	LS	LF

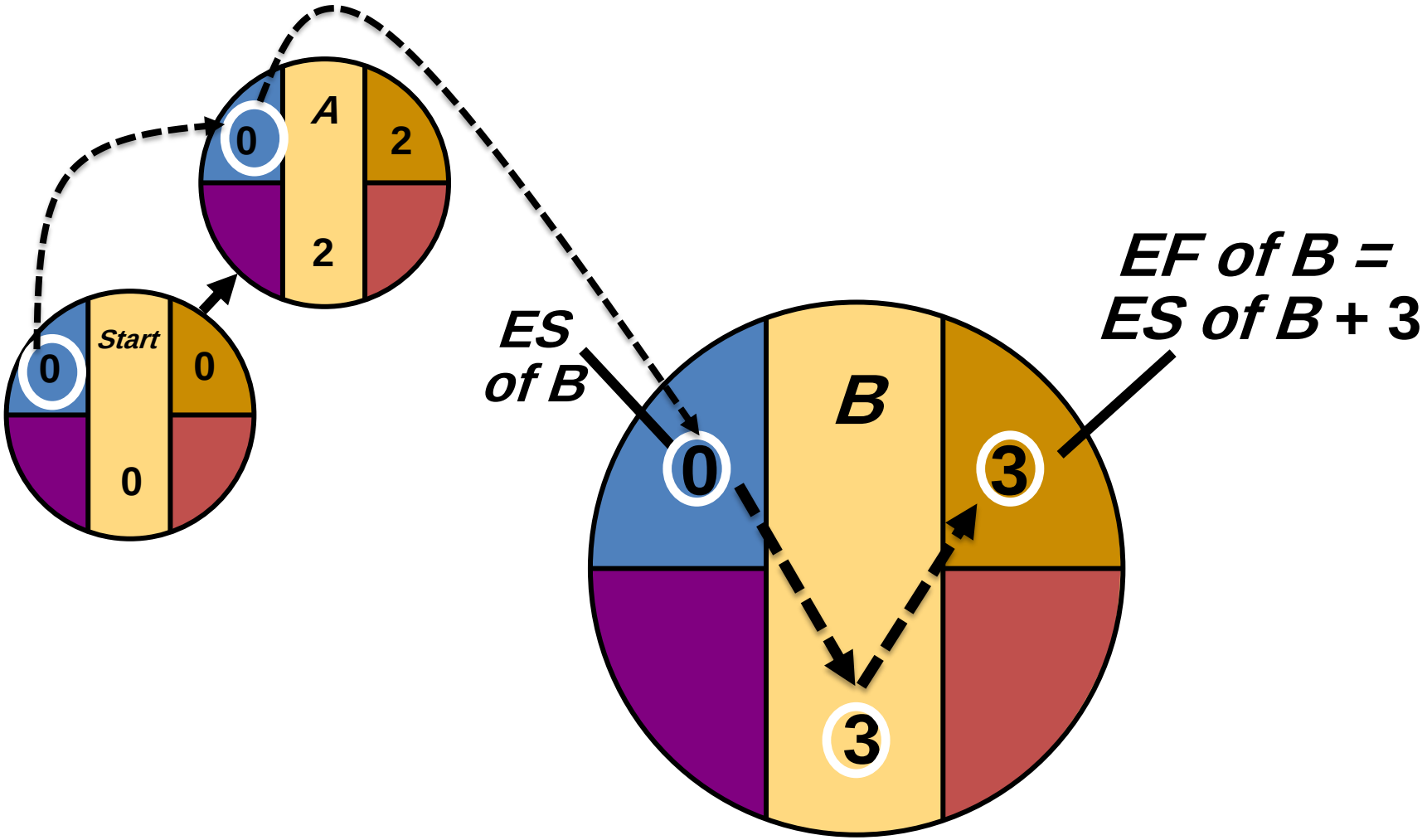
Circle OR Rectangle



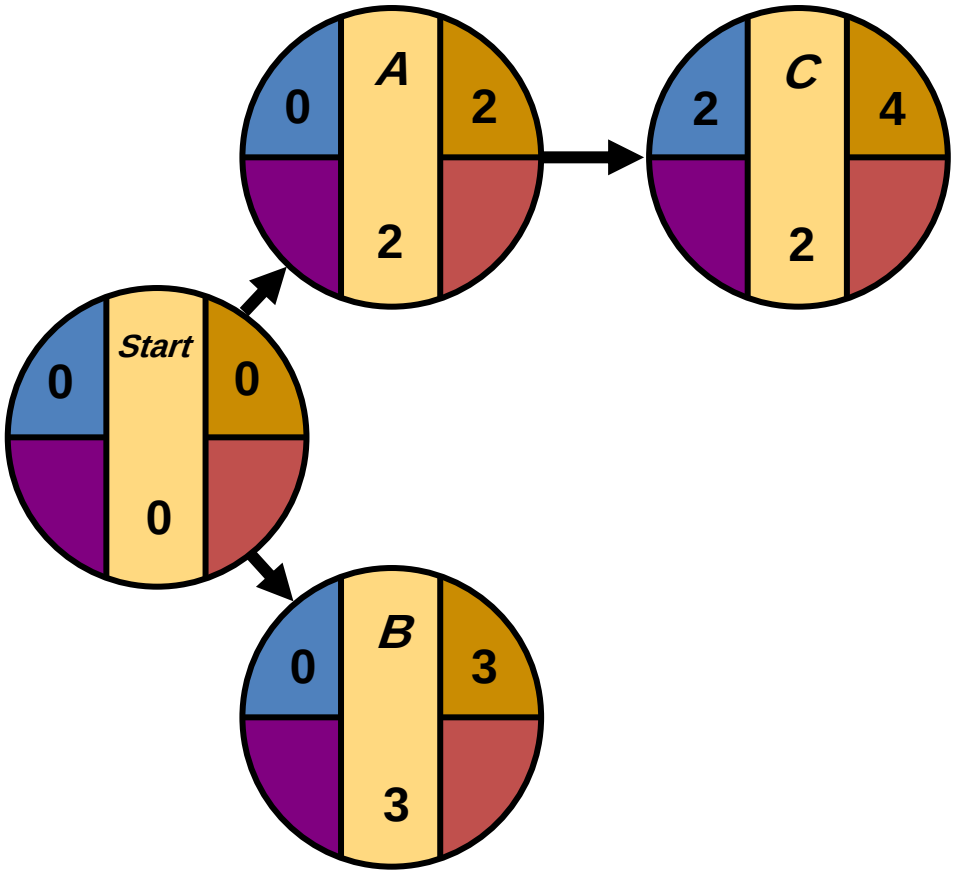
ES/EF Network (*Forward pass*)



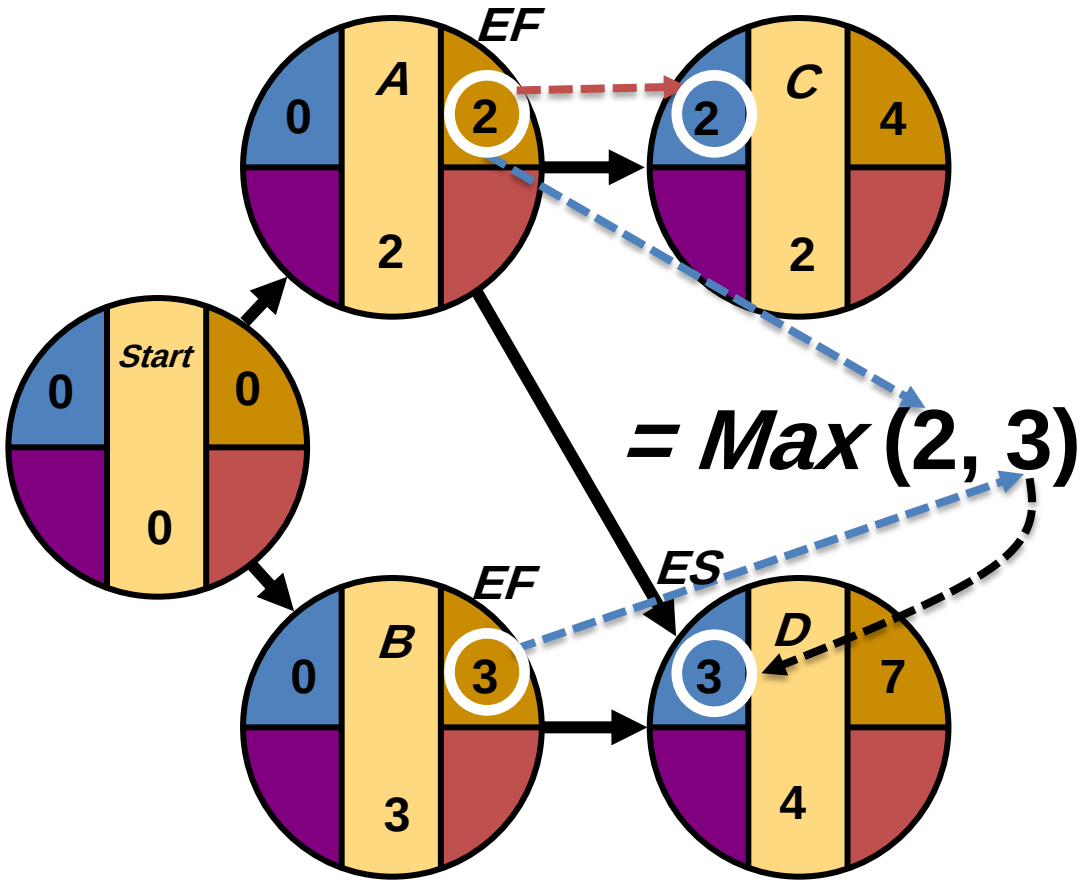
ES/EF Network



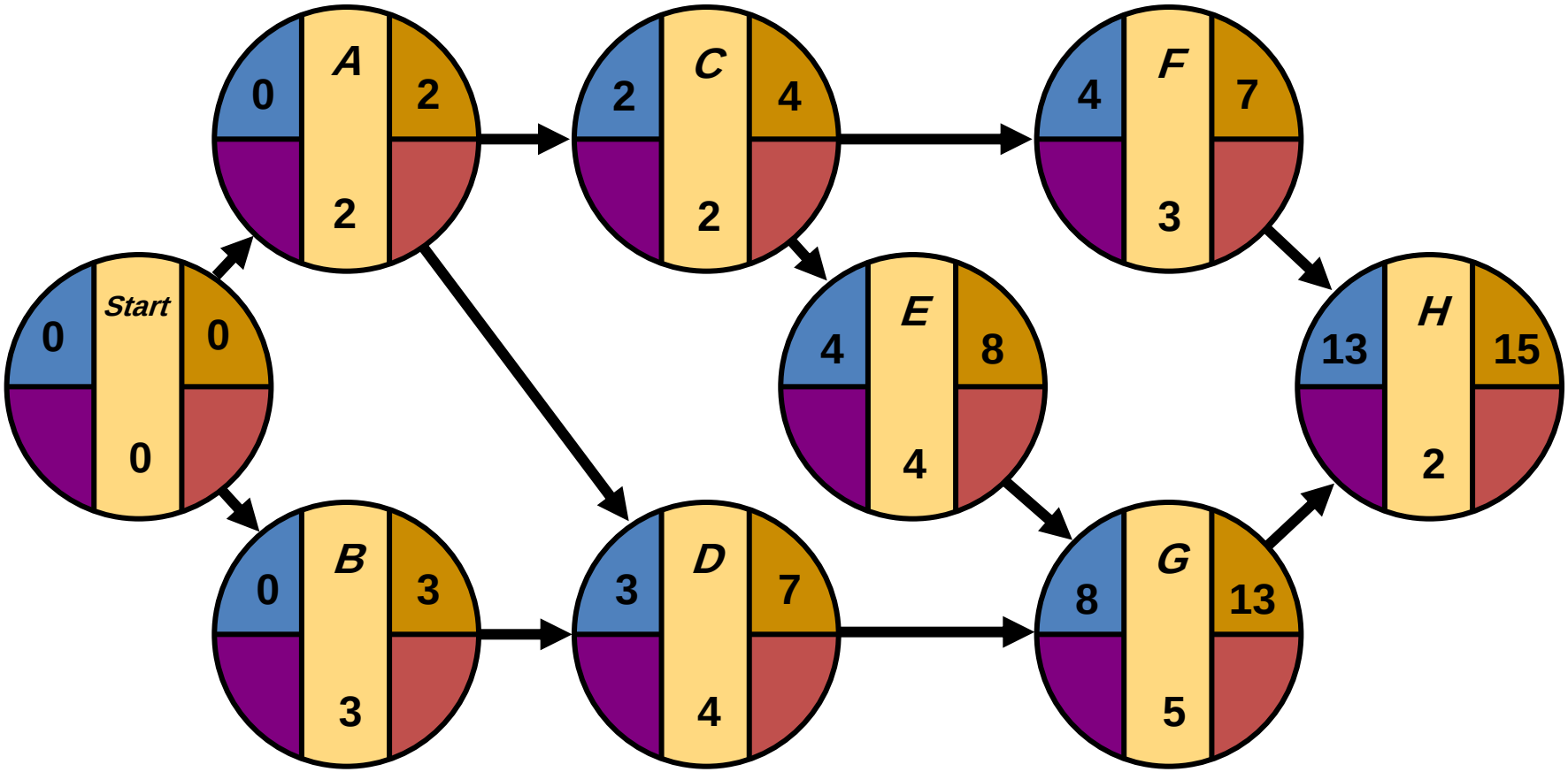
ES/EF Network



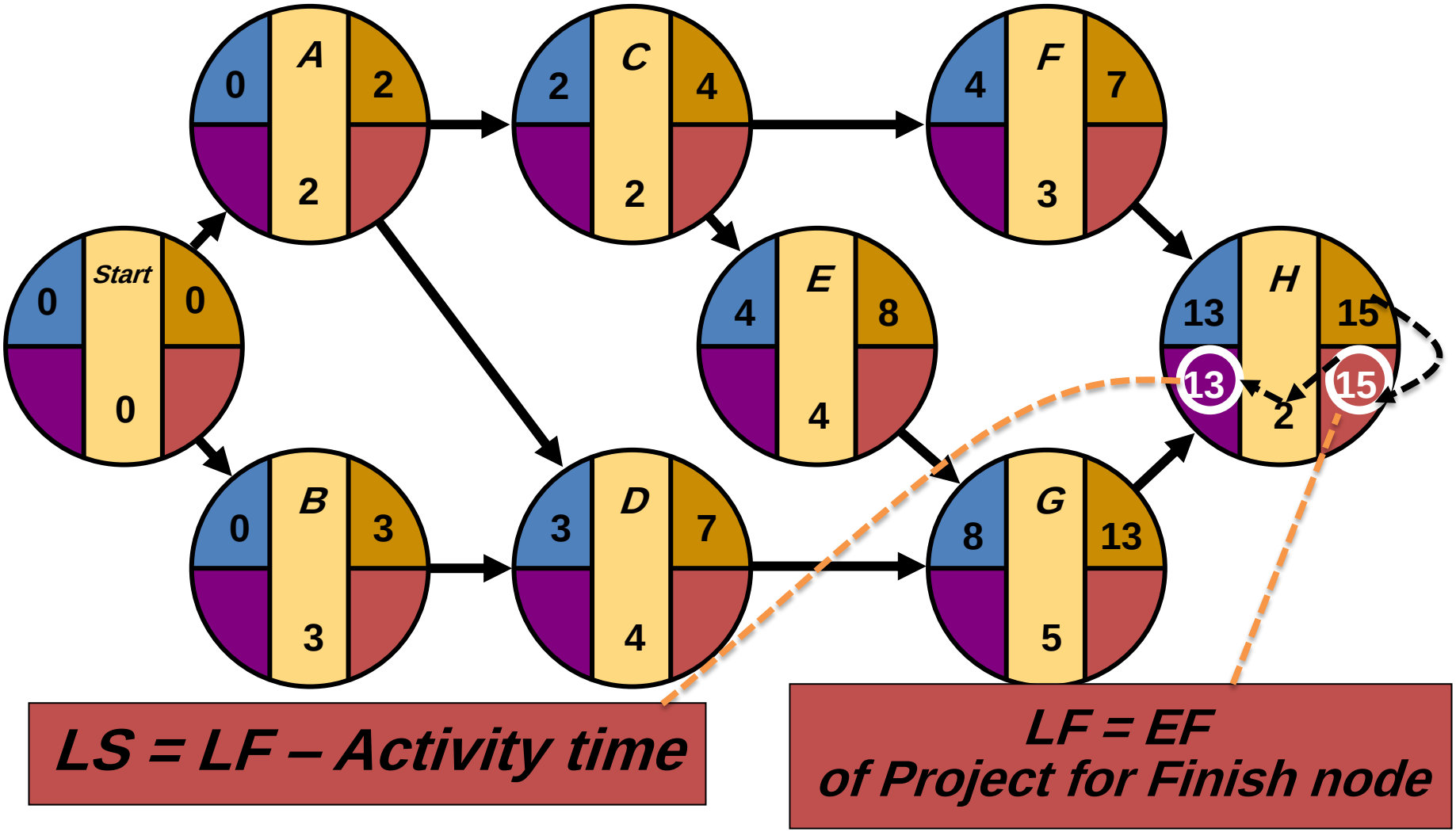
ES/EF Network



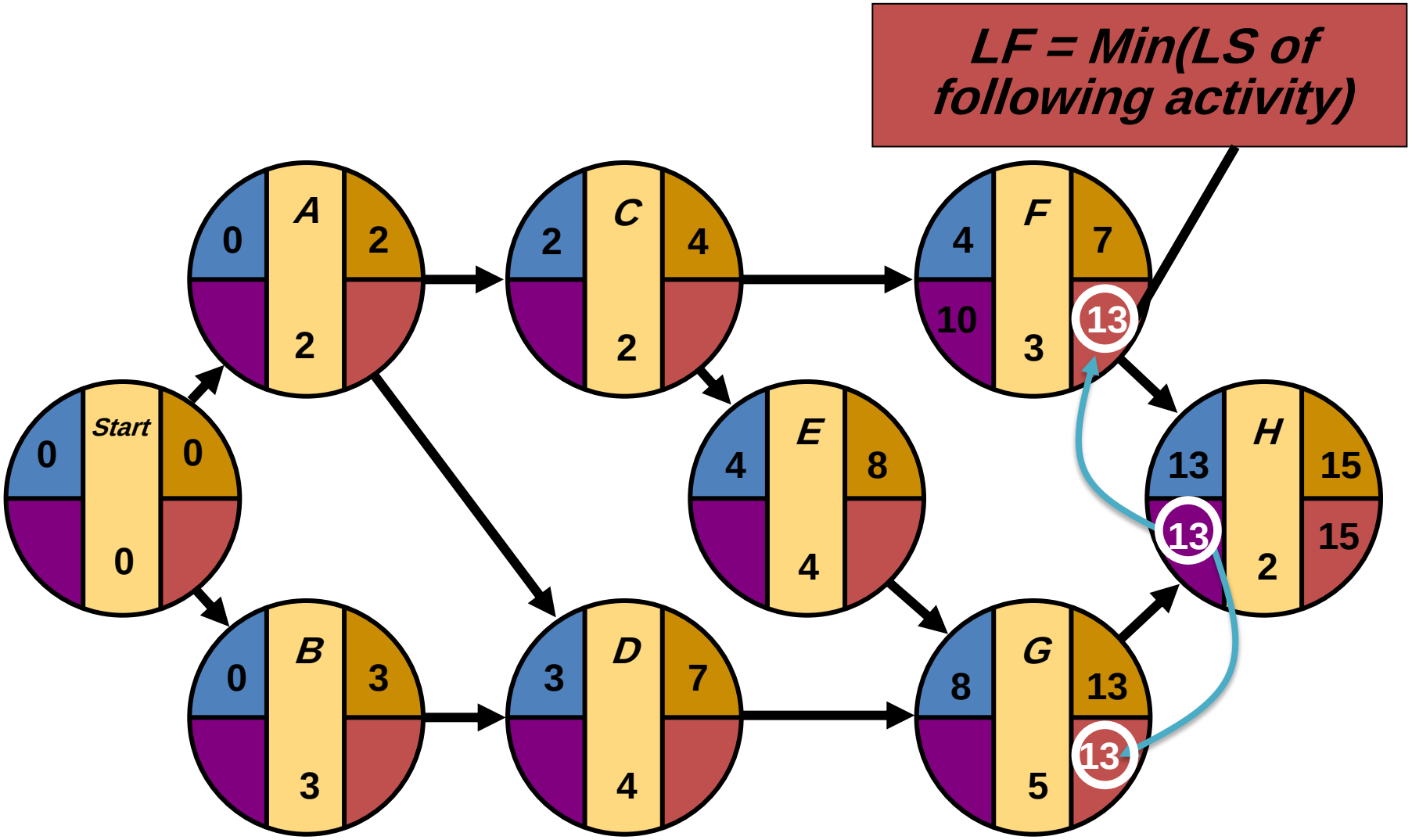
ES/EF Network



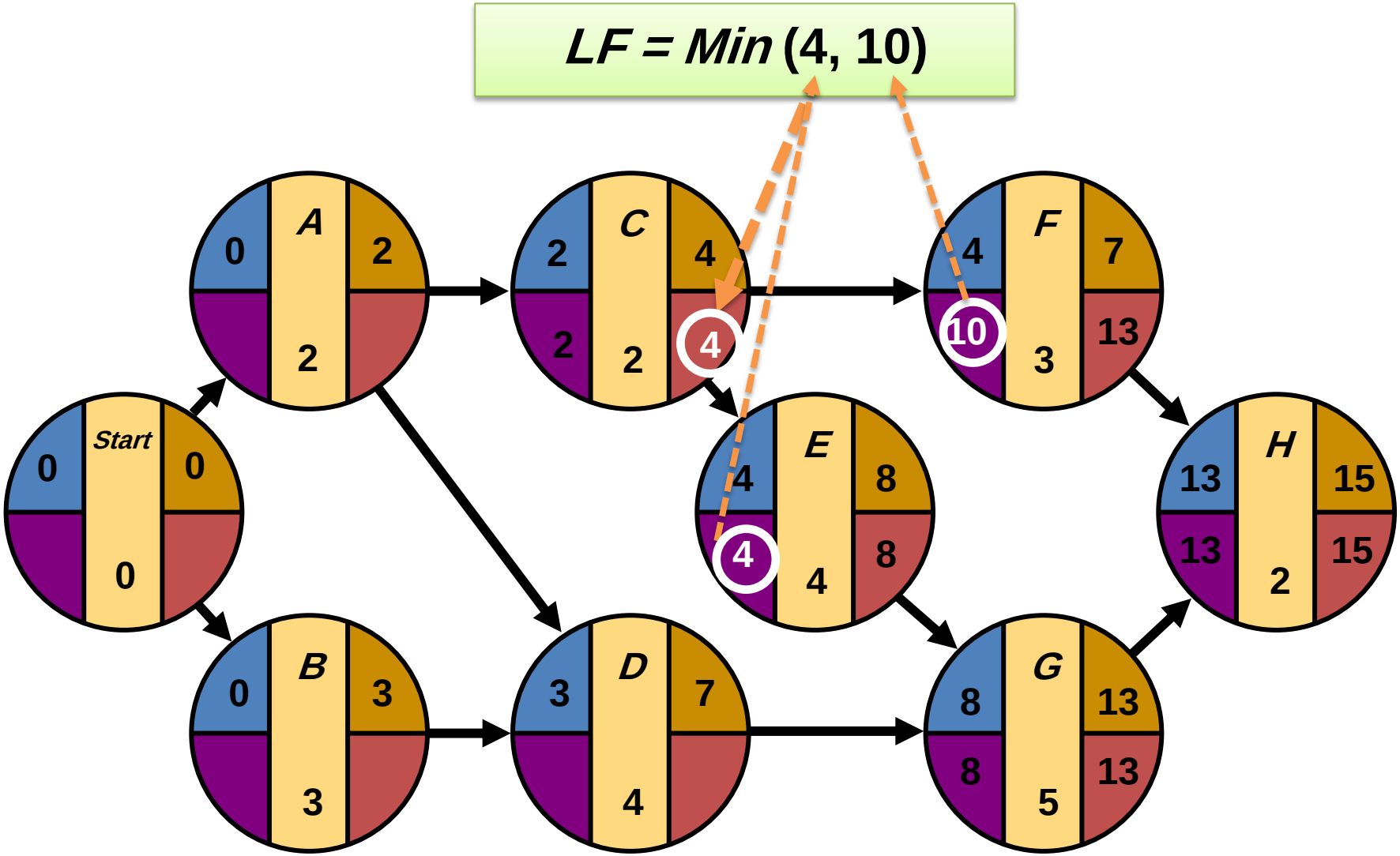
LS/LF Times (*Backward pass*)



LS/LF Times


LF = Min(LS of following activity)
Start
0
0
0
A
2
2
0
2
B
3
3
2
C
4
2
3
D
7
4
4
E
8
4
4
F
7
3
10
13
8
G
13
5
13
13
H
15
2
13
15

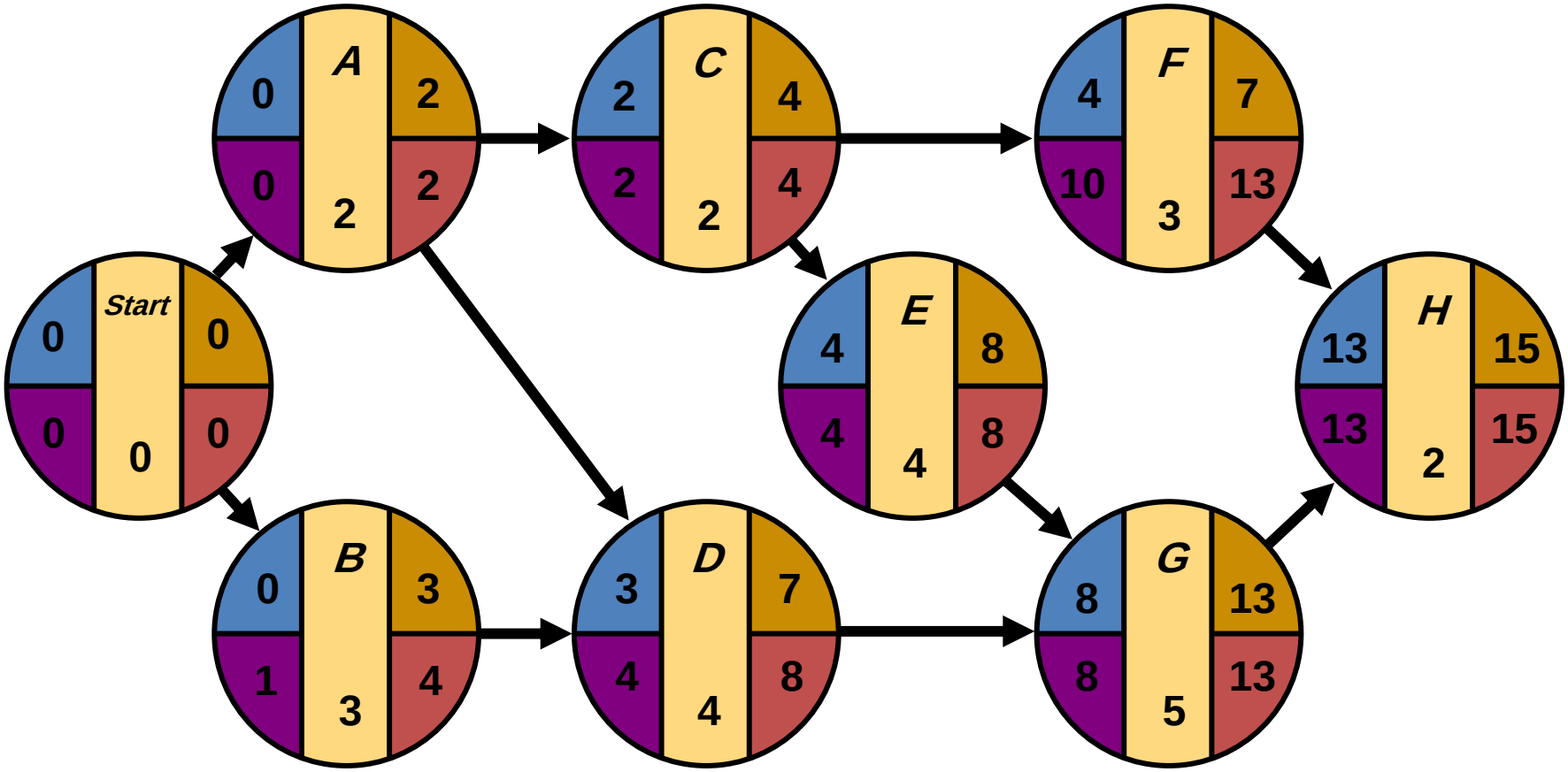
LS/LF Times


LF = Min (4, 10)
Start
0
0
0
A
2
2
0
B
3
3
2
C
4
2
2
3
D
7
4
4
E
8
4
4
4
F
7
3
4
10
G
13
5
8
8
H
15
2
13
13

LS/LF Times

- Early Start
- Early Finish

Forward Pass



Backward Pass

- Late Start
- Late Finish

Computing Slack Time

After computing the ES, EF, LS, and LF times for all activities, compute the slack or free time for each activity

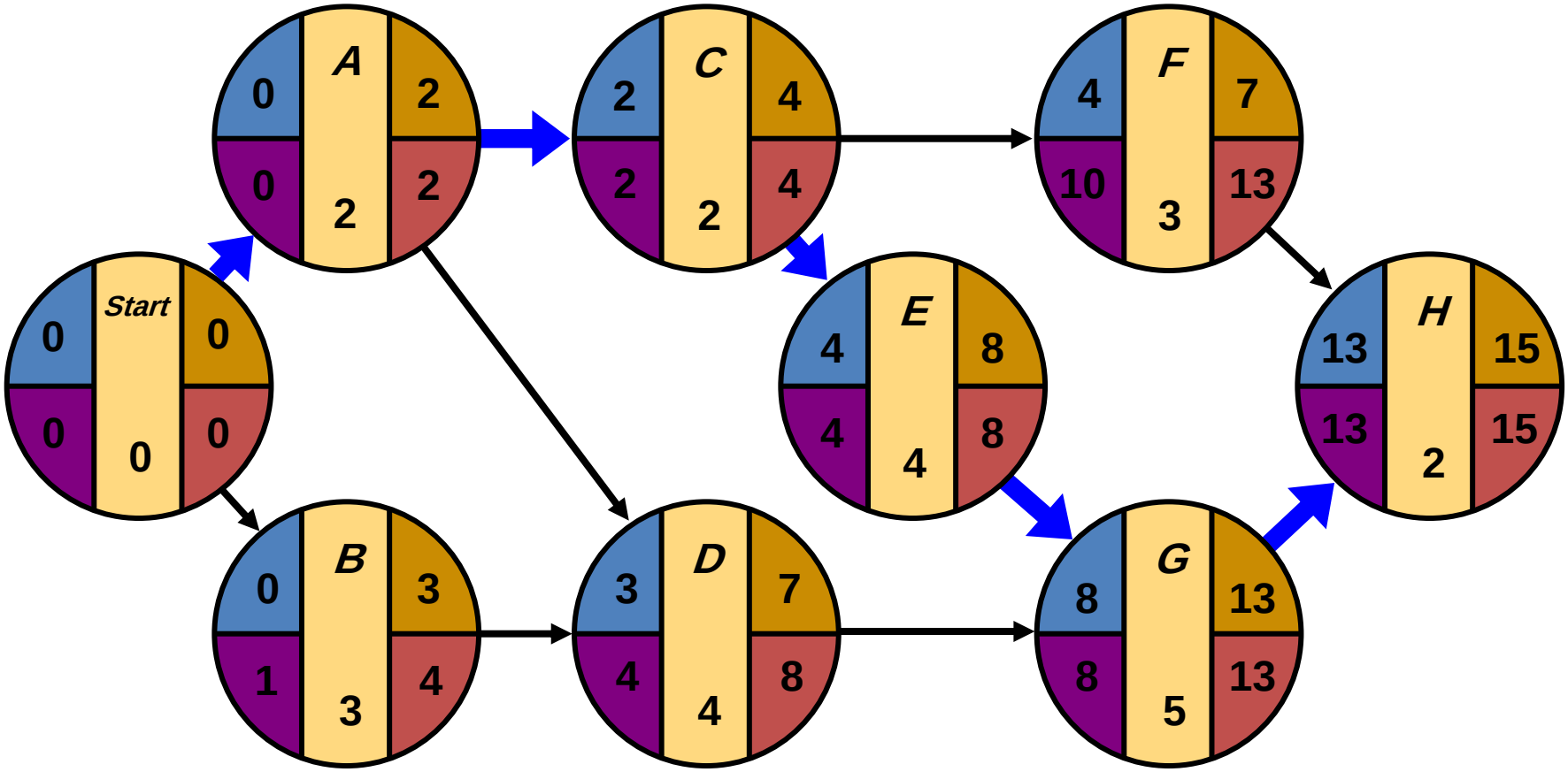
- ☑ ***Slack is the length of time an activity can be delayed without delaying the entire project***

Slack = LS – ES or Slack = LF – EF

Computing Slack Time

Activity	Earliest Start ES	Earliest Finish EF	Latest Start LS	Latest Finish LF	Slack LS – ES	On Critical Path
A	0	2	0	2	0	Yes
B	0	3	1	4	1	No
C	2	4	2	4	0	Yes
D	3	7	4	8	1	No
E	4	8	4	8	0	Yes
F	4	7	10	13	6	No
G	8	13	8	13	0	Yes
H	13	15	13	15	0	Yes

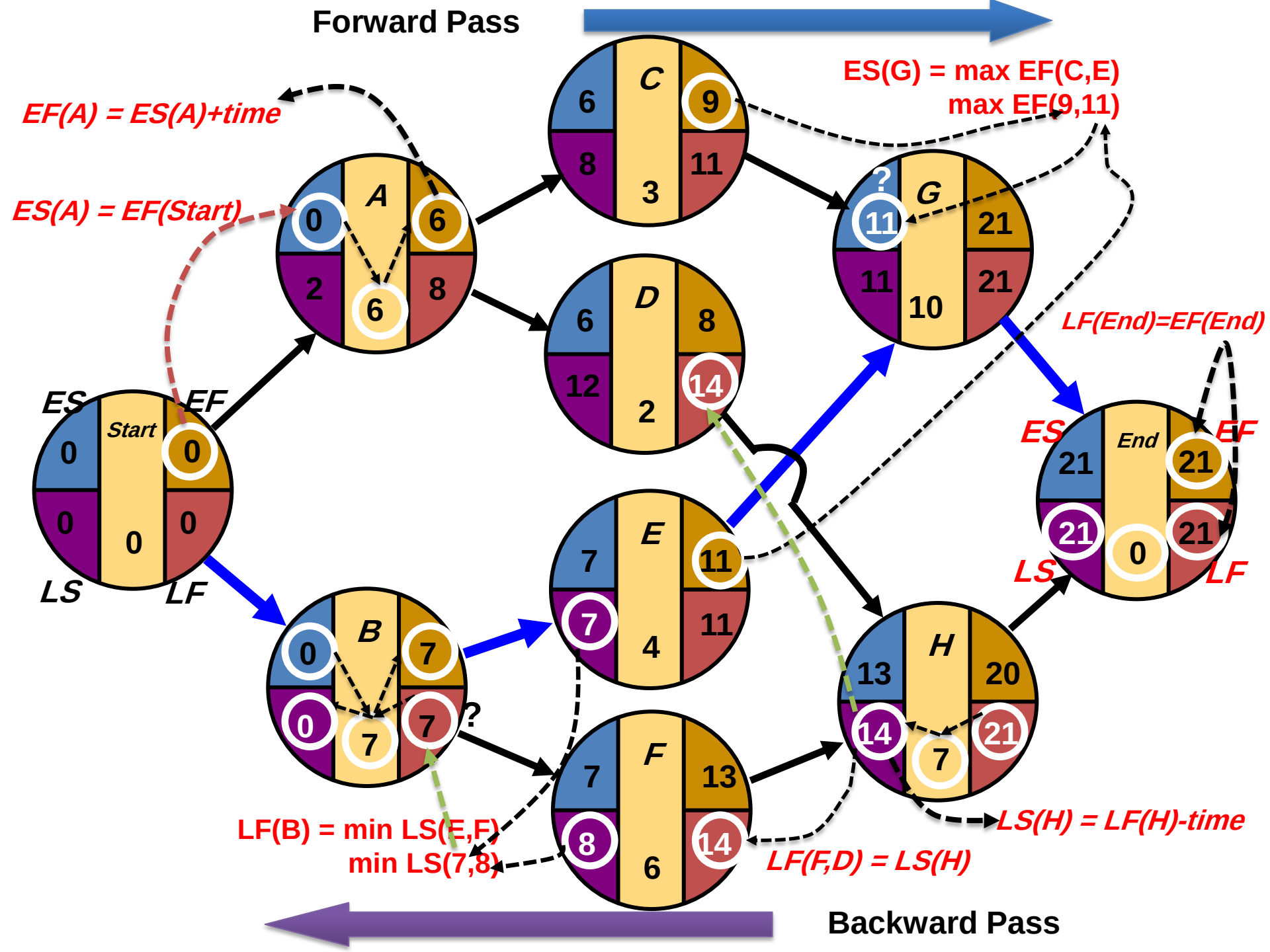
Critical Path



Example(2)

Perform a Critical Path Analysis

Activity	Immediate Predecessors	Time (weeks)
A	-	6
B	-	7
C	A	3
D	A	2
E	B	4
F	B	6
G	C, E	10
H	D, F	7


Forward Pass
EF(A) = ES(A)+time
ES(A) = EF(Start)
ES(G) = max EF(C,E)
max EF(9,11)
LF(End)=EF(End)
ES
EF
Start
0
0
0
0
0
LS
LF
A
0
6
2
6
8
C
6
9
8
3
11
D
6
8
12
2
14
?
G
11
21
11
10
21
ES
EF
End
21
21
21
0
21
LS
LF
E
7
11
7
4
11
B
0
7
0
7
7
?
H
13
20
14
7
21
F
7
13
8
6
14
LF(B) = min LS(E,F)
min LS(7,8)
LF(F,D) = LS(H)
LS(H) = LF(H)-time
Backward Pass

Computing Slack Time

Activity	Earliest Start ES	Earliest Finish EF	Latest Start LS	Latest Finish LF	Slack LS – ES	On Critical Path
A	0	6	2	8	2	No
B	0	7	0	7	0	Yes
C	6	9	8	11	2	No
D	6	8	12	14	6	No
E	7	11	7	11	0	Yes
F	7	13	8	14	1	No
G	11	21	11	21	0	Yes
H	13	20	14	21	1	No

The Critical Path is B-E-G

Project Completion Time is: 21

Program Evaluation & Review Techniques (PERT)

Learning Objectives

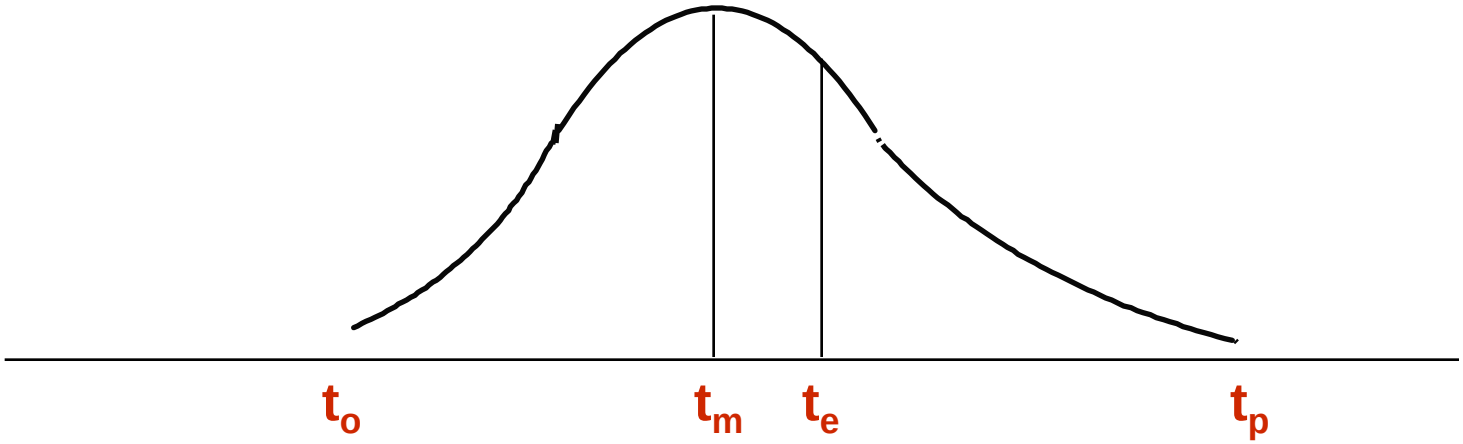
- Give a general description of PERT/CPM techniques.
- Construct simple network diagrams.
- List the kinds of information that a PERT or CPM analysis can provide.
- Analyze networks with deterministic times.
- Analyze networks with probabilistic times.

Probabilistic Time Estimates

- Optimistic time
  - Time required under optimal conditions
- Pessimistic time
  - Time required under worst conditions
- Most likely time
  - Most probable length of time that will be required

Probabilistic Estimates

Beta Distribution



Activity start

Optimistic time

Most likely time (mode)

Pessimistic time

Expected Time

$$t_e = \frac{t_o + 4t_m + t_p}{6}$$

t_e = expected time
t_o = optimistic time
t_m = most likely time
t_p = pessimistic time

Variance

$$\sigma^2 = \frac{(t_p - t_o)^2}{36}$$

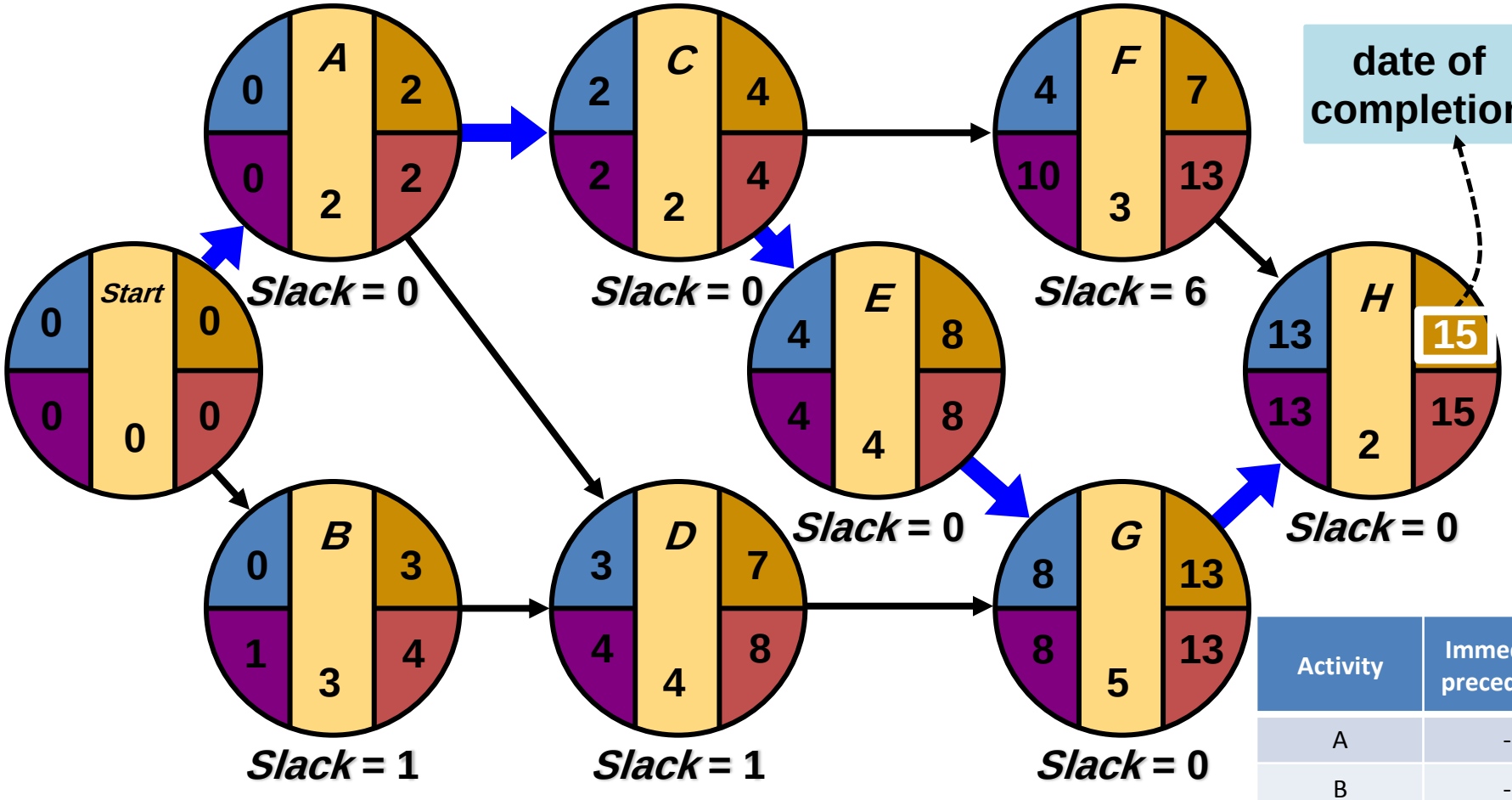
σ^2 = variance
t_o = optimistic time
t_p = pessimistic time

Computing Variance

Let $t_e = t$, $t_o = a$, $t_m = m$, $t_p = b$

Activity	Optimistic a	Most Likely m	Pessimistic b	Expected Time $t = (a + 4m + b)/6$ (Mean)	Variance $[(b - a)/6]^2$	Immediate Pred.
A	1	2	3	2	0.11	—
B	2	3	4	3	0.11	—
C	1	2	3	2	0.11	A
D	2	4	6	4	0.44	A, B
E	1	4	7	4	1.00	C
F	1	2	9	3	1.78	C
G	3	4	11	5	1.78	D, E
H	1	2	3	2	0.11	F, G

Critical Path and Slack Times



Activity	Immediate precedence
A	-
B	-
C	A
D	A,B
E	C
F	C
G	D,E
H	F,G

Probability of Project Completion

Project variance is computed by summing the variances of critical activities

σ_p^2 = *Project variance*

= $\sum$(*variances of activities on critical path*)

Probability of Project Completion

Project variance is computed by summing the variances of critical activities

Project variance on critical path..

$$\sigma_p^2 = \sigma_A + \sigma_C + \sigma_E + \sigma_G + \sigma_H$$

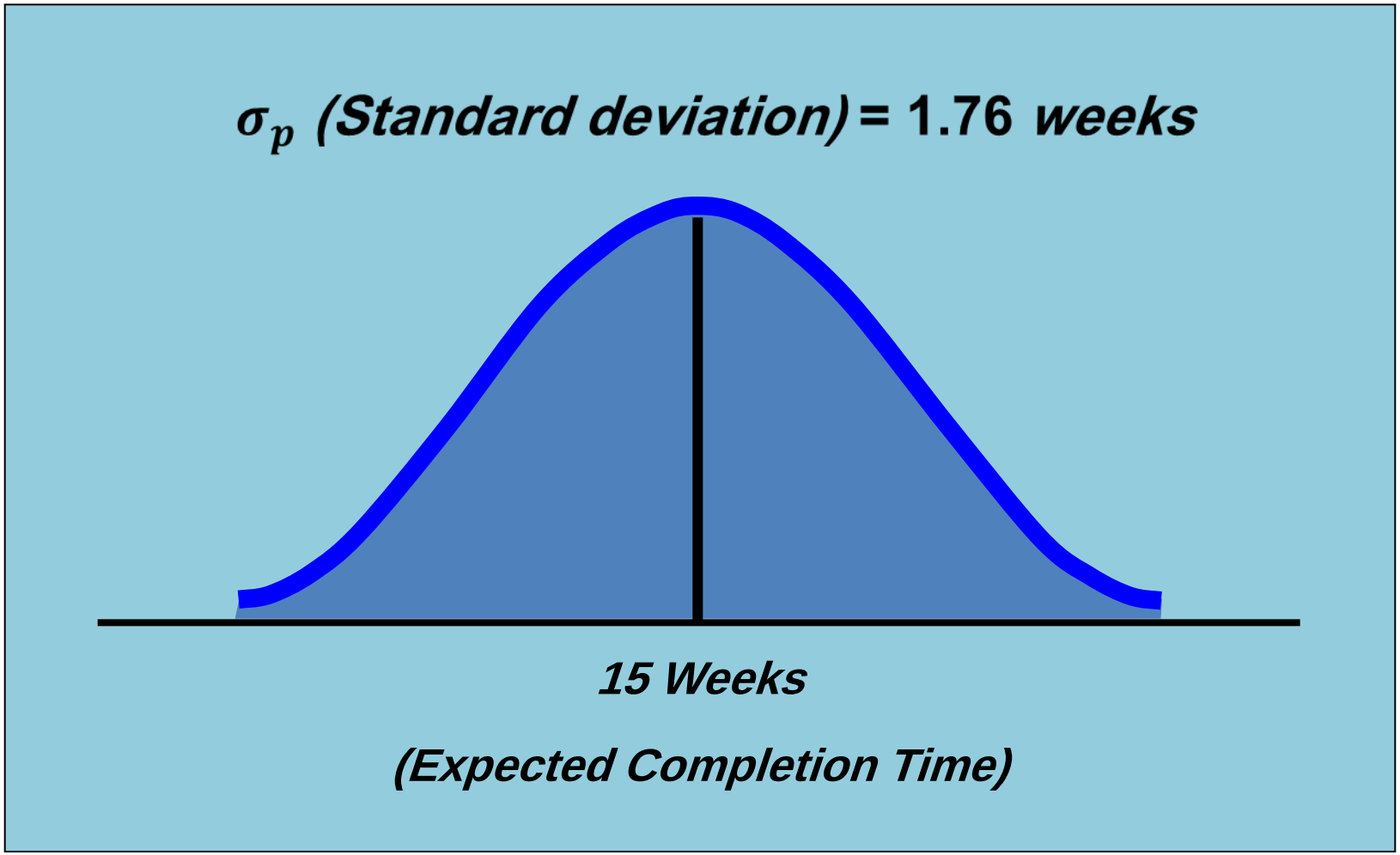
$$\sigma_p^2 = 0.11 + 0.11 + 1.00 + 1.78 + 0.11 = 3.11$$

Project standard deviation

$$\sigma_p = \sqrt{\text{Project variance}}$$

$$= \sqrt{3.11} = 1.76 \text{ weeks}$$

Probability of Project Completion



Probability of Project Completion

What is the probability of completing the project on or before the 16 weeks?

due date = 16 → x
Exp. date = 15 → μ

From the Standard Normal Distribution table: (z table)

$$Z = \begin{pmatrix} \text{due} \\ \text{date} \end{pmatrix} - \begin{pmatrix} \text{expected date} \\ \text{of completion } \mu \end{pmatrix} / \sigma_p$$

$$= (16 \text{ weeks} - 15 \text{ weeks}) / 1.76$$

$$= 0.57$$

0.5 + 0.07

Where Z is the number of standard deviations the due date lies from the mean

$P(x \leq 16) = P[x - \mu / \sigma \leq 16-15/1.76]$

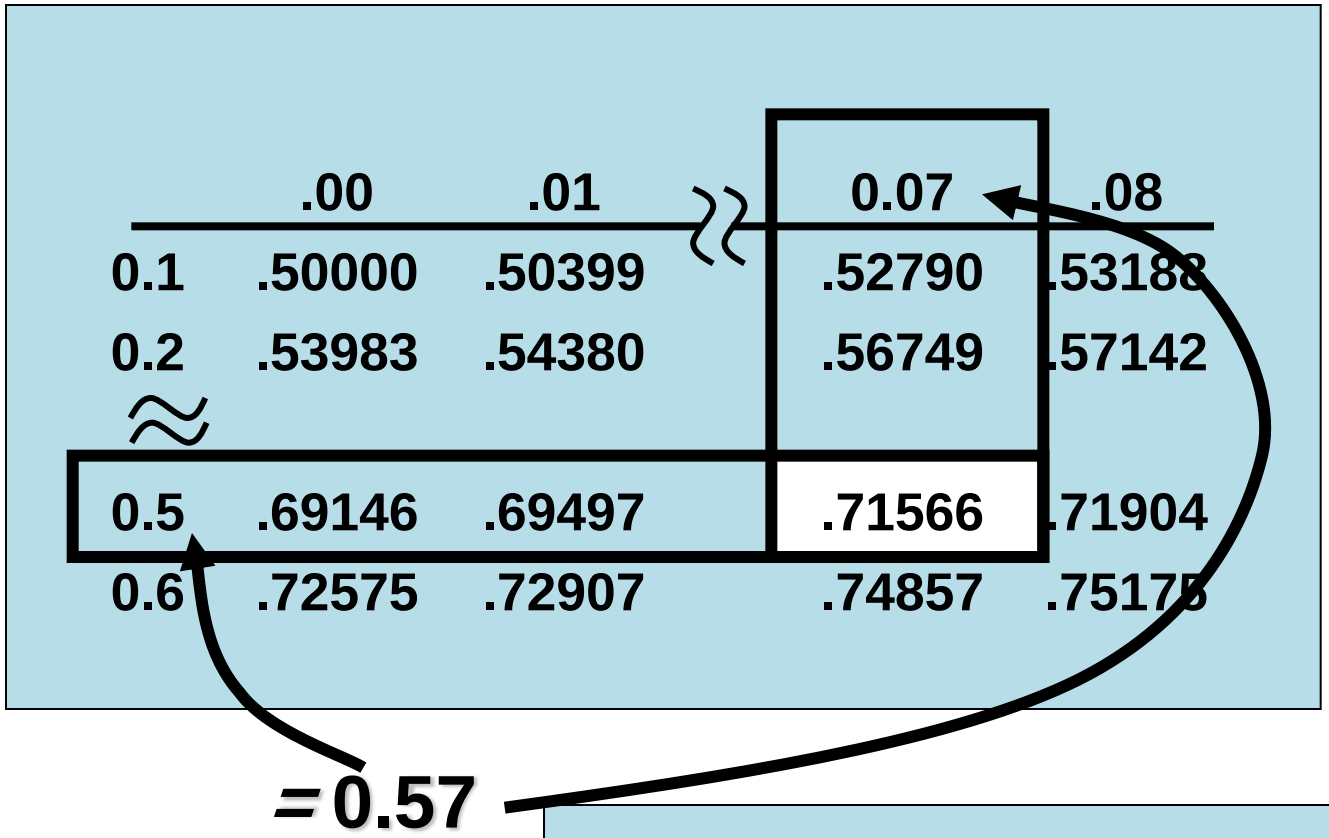
$= P[z \leq 0.57] =$ 0.7157 This value is obtain from Standard Normal Distribution table, therefore the probability of completing the project on or before 16 weeks is 0.7157 , i.e 71.57%

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
+0	.50000	.50399	.50798	.51197	.51595	.51994	.52392		.53188	.53586
+0.1	.53983	.54380	.54776	.55172	.55567	.55966	.56360	.56749	.57142	.57535
+0.2	.57926	.58317	.58706	.59095	.59483	.59871	.60257	.60642	.61026	.61409
+0.3	.61791	.62172	.62552	.62930	.63307	.63683	.64058	.64431	.64803	.65173
+0.4	.65542	.65910	.66276	.66640	.67003	.67364	.67724	.68082	.68439	.68793
+0.5	.69146	.69497	.69847	.70194	.70540	.70884	.71226	.71566	.71904	.72240
+0.6	.72575	.72907	.73237	.73565	.73891	.74215	.74537	.74857	.75175	.75490
+0.7	.75804	.76115	.76424	.76730	.77035	.77337	.77637	.77935	.78230	.78524
+0.8	.78814	.79103	.79389	.79673	.79955	.80234	.80511	.80785	.81057	.81327
+0.9	.81594	.81859	.82121	.82381	.82639	.82894	.83147	.83398	.83646	.83891
+1	.84134	.84375	.84614	.84849	.85083	.85314	.85543	.85769	.85993	.86214
+1.1	.86433	.86650	.86864	.87076	.87286	.87493	.87698	.87900	.88100	.88298
+1.2	.88493	.88686	.88877	.89065	.89251	.89435	.89617	.89796	.89973	.90147
+1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91308	.91466	.91621	.91774
+1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
+1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
+1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
+1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
+1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
+1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
+2	.97725	.97778	.97					.98077	.98124	.98169
+2.1	.98214	.98257	.98					.98500	.98537	.98574
+2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
+2.3	.98928	.98956	.98983	.99010	.99036	.99061	.99086	.99111	.99134	.99158
+2.4	.99180	.99202	.99224	.99245	.99266	.99286	.99305	.99324	.99343	.99361
+2.5	.99379	.99396	.99413	.99430	.99446	.99461	.99477	.99492	.99506	.99520
+2.6	.99534	.99547	.99560	.99573	.99585	.99598	.99609	.99621	.99632	.99643
+2.7	.99653	.99664	.99674	.99683	.99693	.99702	.99711	.99720	.99728	.99736
+2.8	.99744	.99752	.99760	.99767	.99774	.99781	.99788	.99795	.99801	.99807
+2.9	.99813	.99819	.99825	.99831	.99836	.99841	.99846	.99851	.99856	.99861
+3	.99865	.99869	.99874	.99878	.99882	.99886	.99889	.99893	.99896	.99900
+3.1	.99903	.99906	.99910	.99913	.99916	.99918	.99921	.99924	.99926	.99929
+3.2	.99931	.99934	.99936	.99938	.99940	.99942	.99944	.99946	.99948	.99950
+3.3	.99952	.99953	.99955	.99957	.99958	.99960	.99961	.99962	.99964	.99965
+3.4	.99966	.99968	.99969	.99970	.99971	.99972	.99973	.99974	.99975	.99976
+3.5	.99977	.99978	.99978	.99979	.99980	.99981	.99981	.99982	.99983	.99983
+3.6	.99984	.99985	.99985	.99986	.99986	.99987	.99987	.99988	.99988	.99989
+3.7	.99989	.99990	.99990	.99990	.99991	.99991	.99992	.99992	.99992	.99992
+3.8	.99993	.99993	.99993	.99994	.99994	.99994	.99994	.99995	.99995	.99995
+3.9	.99995	.99995	.99996	.99996	.99996	.99996	.99996	.99996	.99997	.99997
+4	.99997	.99997	.99997	.99997	.99997	.99997	.99998	.99998	.99998	.99998

Z TABLE-Positive Z score Table

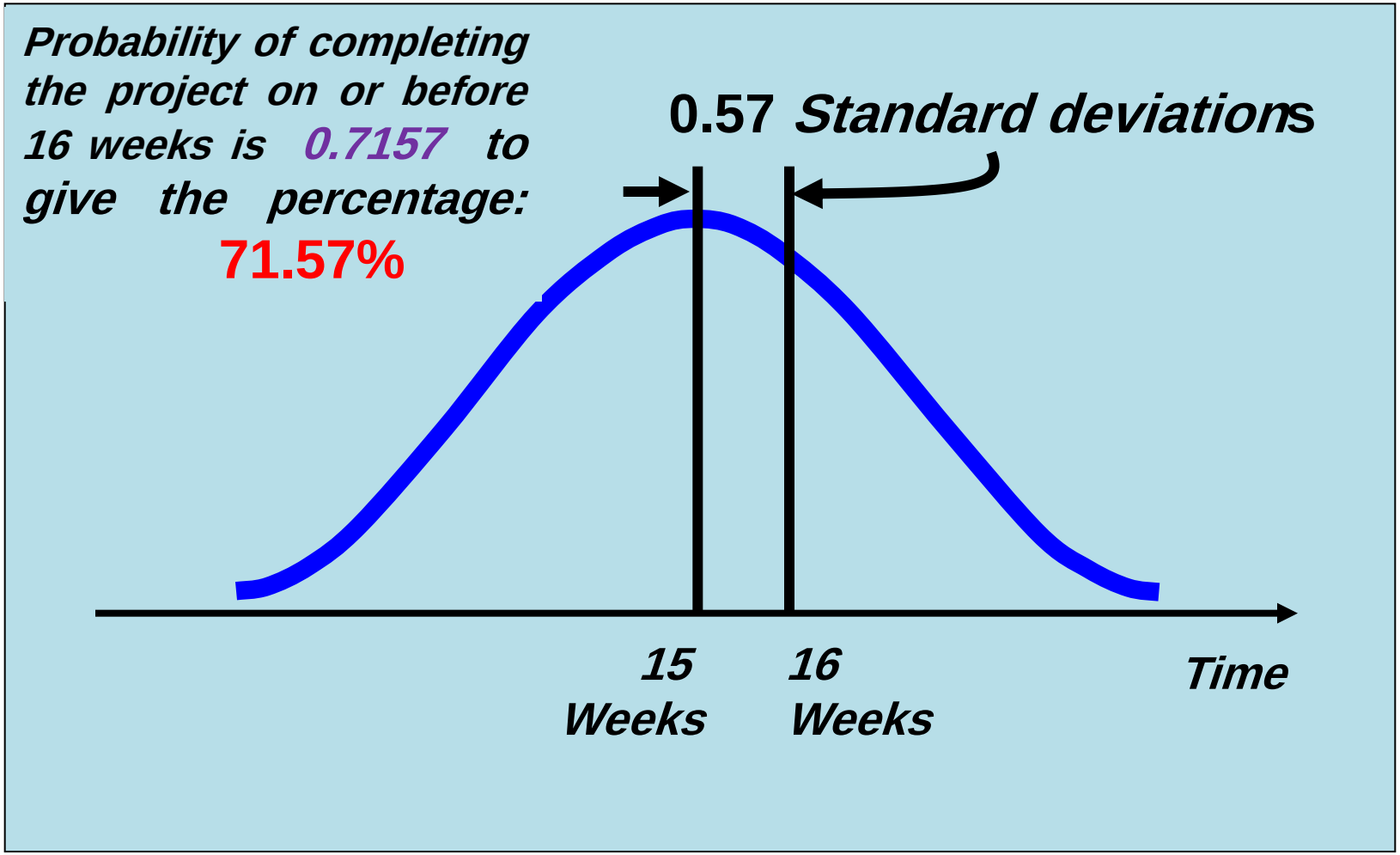
Probability of Project Completion

What is the probability this project can be completed on or before the 16 week deadline?



Where Z is the number of standard deviations the due date lies from the mean

Probability of Project Completion



Q: If we have the table below, by using PERT method, answer the following?

Activity	Predecessors	Duration (weeks)		
		a	***m***	***b***
A	-	5	6	7
B	-	1	3	5
C	-	1	4	7
D	A	1	2	3
E	B	1	2	9
F	C	1	5	9
G	C	2	2	8
H	E,F	4	4	10
I	D	2	5	8
J	H,G	2	2	8

1. **Construct the project network.**
2. **Find the expected duration and variance of each activity.**
3. **Determine the critical path and expected project completion time.**
4. **What is the probability of completing the project on or before 20 weeks.**

Chapter 6- Software Quality Management

جامعة القادسية/ كلية علوم الحاسوب وتكنولوجيا المعلومات

/ قسم نظم المعلومات الحاسوبية/ المرحلة الثالثة

What is Quality Management?

Managing the quality of the software process and products

Objectives

- To introduce the quality management process and key quality management activities
- To explain the role of standards in quality management
- To explain the relationship between quality attributes and software metrics
- To explain how measurement may be used in assessing software quality

Software quality management

- Concerned with ensuring that the required level of quality is achieved in a software product
- Involves defining appropriate quality standards and procedures and ensuring that these are followed
- Should aim to develop a ‘quality culture’ where quality is seen as everyone’s responsibility

What is quality?

- Quality, simplistically, means that a product should meet its specification
- This is problematical for software systems
  - Tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.)
  - Some quality requirements are difficult to specify in an unambiguous way
  - Software specifications are usually incomplete and often inconsistent

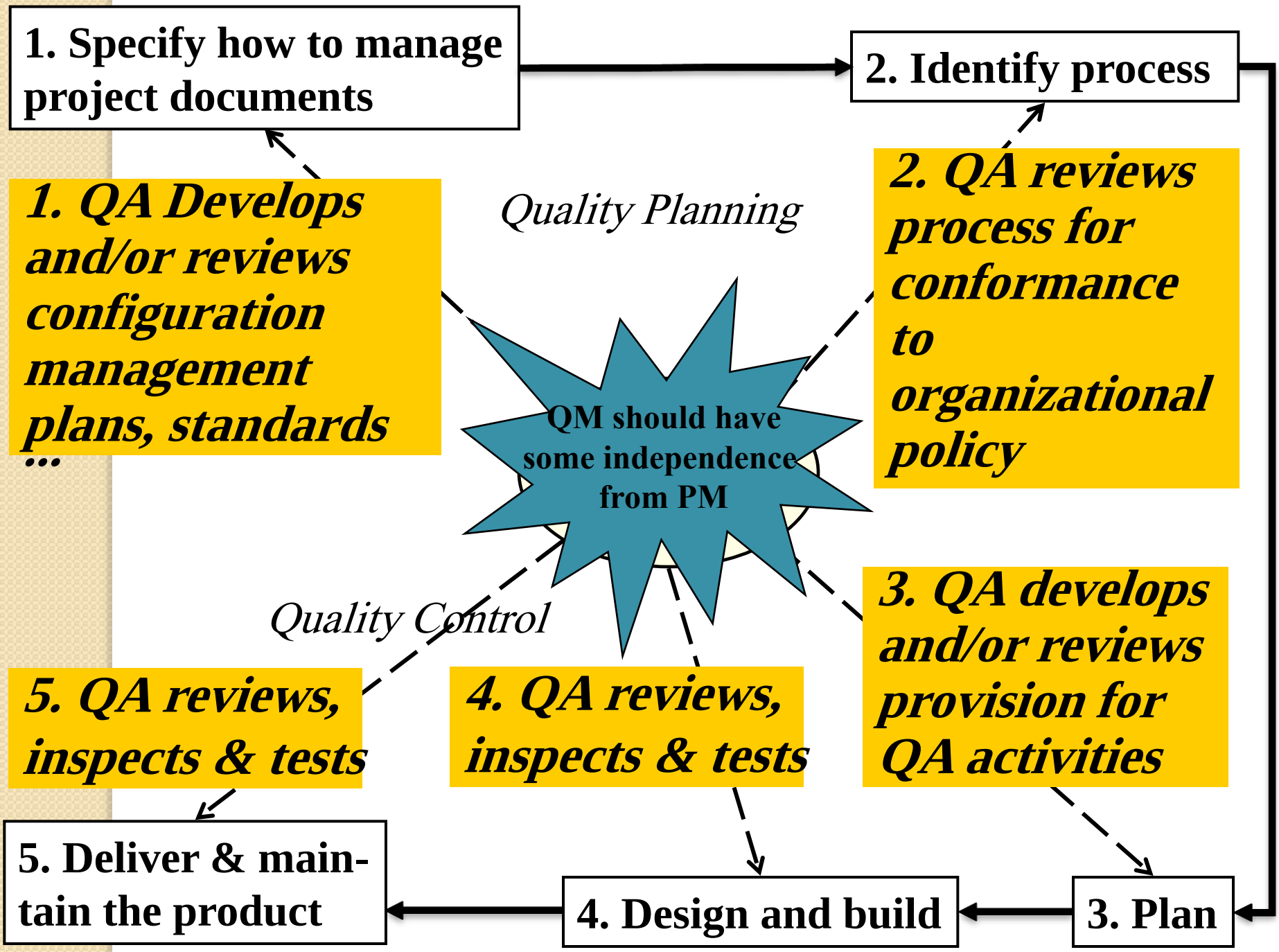
Software quality attributes

A high quality software product …

- Satisfies clearly stated requirements
- Check its inputs and that it reacts in predictable ways to illegal inputs
- Has been inspected thoroughly by others
- Has been tested exhaustively by others
- Is thoroughly documented
- Has a known defect rate

The quality compromise

- We cannot wait for specifications to improve before paying attention to quality management
- Must put procedures into place to improve quality in spite of imperfect specification
- Quality management is therefore not just concerned with reducing defects but also with other product qualities


1. Specify how to manage project documents
2. Identify process
1. QA Develops and/or reviews configuration management plans, standards …
Quality Planning
2. QA reviews process for conformance to organizational policy
QM should have some independence from PM
Quality Control
3. QA develops and/or reviews provision for QA activities
5. QA reviews, inspects & tests
4. QA reviews, inspects & tests
5. Deliver & main-tain the product
4. Design and build
3. Plan

Quality assurance and standards

- Standards are the key to effective quality management
- They may be international, national, organizational or project standards
- Product standards define characteristics that all components should exhibit e.g. a common programming style
- Process standards define how the software process should be enacted

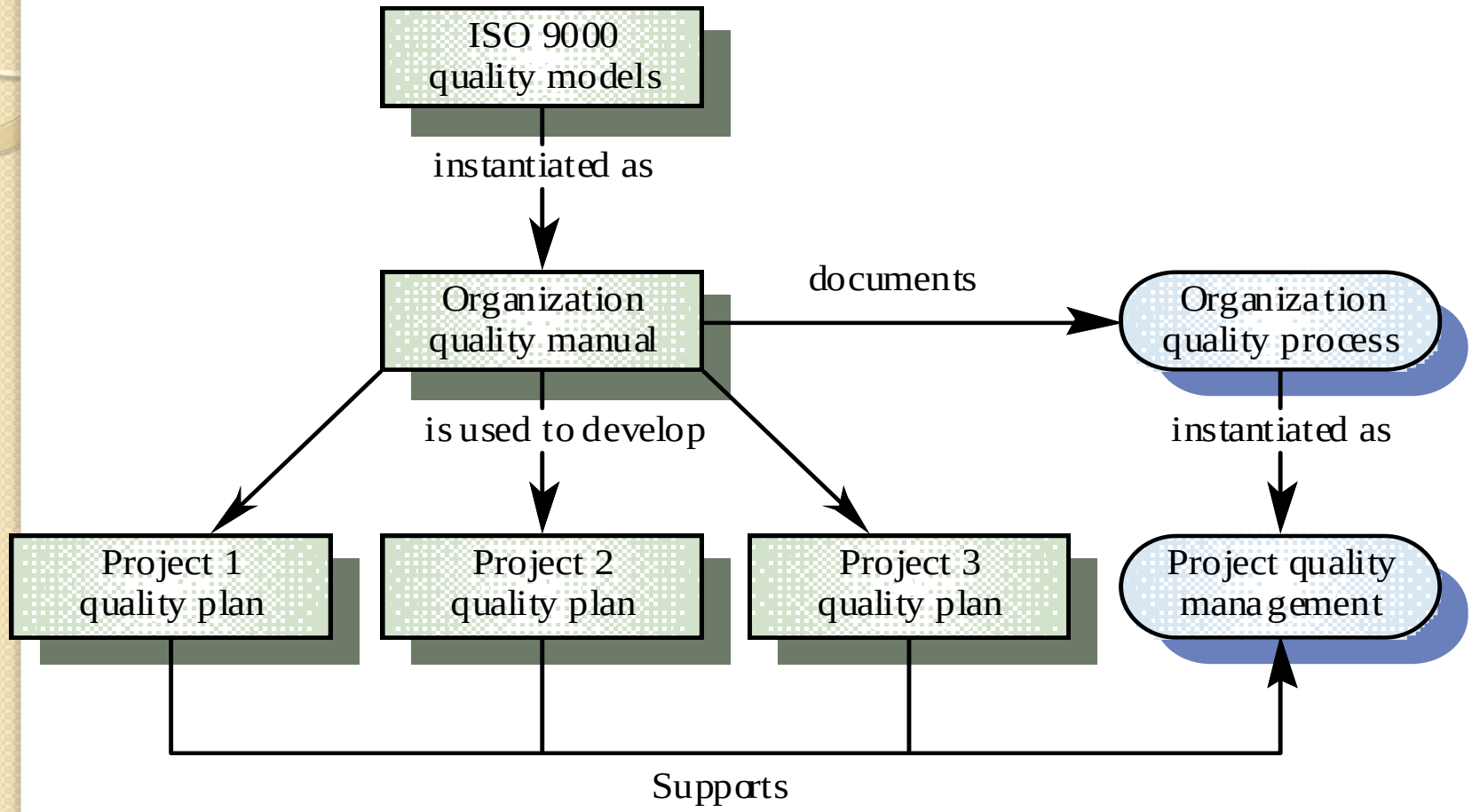
Importance of standards

- Encapsulation of best practice- avoids repetition of past mistakes
- Framework for quality assurance process – it involves checking standard compliance
- Provide continuity - new staff can understand the organisation by understand the standards applied

ISO 9000

- ***International Standard Organization*** for quality management
- Applicable to a range of organisations from manufacturing to service industries
- ISO 9001 applicable to organisations which design, develop and maintain products
- ISO 9001 is a generic model of the quality process that must be instantiated for each organisation

ISO 9000 and quality management



Problems with standards

- Not seen as relevant and up-to-date by software engineers
- Involve too much bureaucratic form filling
- Unsupported by software tools so tedious manual work is involved to maintain standards