Unit-3
Software Project Management

What is project

- A project is a group of tasks that need to complete to reach a clear result.
- A project is also defines as a set of inputs and outputs which are required to achieve a goal.
- Projects can vary from simple to difficult and can be operated by one person or a hundred.

Software project Management

- Software project management is an art and discipline of planning and supervising software projects.
- It is a sub-discipline of software project management in which software projects planned, implemented, monitored and controlled.
- It is a procedure of managing, allocating and timing resources to develop computer software that fulfills requirements.
- In software Project Management, the client and the developers need to know the length, period and cost of the project.

Perquisite of software project management

- There are three needs for software project management. These are:
  - Time
  - Cost
  - Quality
- It is an essential part of the software organization to deliver a quality product, keeping the cost within the clients budget and deliver the project as per schedule. There are various factors, both external and internal, which may impact the triple factor. Any of three- factor can severely affect the other two.

Types of Management in Software Project Management:

1. Conflict Management
2. Risk Management
3. Requirement Management
4. Change Management
5. Software Configuration Management
6. Release Management

1. Conflict Management

- Conflict management is the process to restrict the negative features of conflict while increasing the positive features of conflict. The goal of conflict management is to improve learning and group results including efficacy or performance in an organization setting.

2. Risk Management

- Risk management is the analysis and identification of risks that is followed by synchronized and economical implementation of resources to minimize, operate and control the possibility or effect of unfortunate events or to maximize the realization of opportunities.

3. Requirement Management

- It is the process of analyzing, prioritizing, tracking, and documenting requirements and then supervising change and communicating to pertinent(suitable) stakeholders. It is a continuous process during a project.

4. Change Management

- Change management is a systematic approach to dealing with the transition or transformation of an organization's goals, processes, or technologies. The purpose of change management is to execute strategies for effecting change, controlling change, and helping people to adapt to change.

5. Software Configuration Management

- Software configuration management is the process of controlling and tracking changes in the software, part of the larger cross-disciplinary field of configuration management. Software configuration management includes revision control and the inauguration of baselines.

6. Release Management

- Release Management is the task of planning, controlling, and scheduling the built-in deploying releases. Release management ensures that the organization delivers new and enhanced services required by the customer while protecting the integrity of existing services.

Project Manager

- A project manager is a character who has the overall responsibility for the planning, design, execution, monitoring, controlling and closure of a project.
- A project manager represents an essential role in the achievement of the projects.
- A project manager is a character who is responsible for giving decisions, both large and small projects.
- The project manager is used to manage the risk and minimize uncertainty. Every decision the project manager makes must directly profit their project.

Role of Project Manager

1. Leader

- A project manager must lead his team and should provide them direction to make them understand what is expected from all of them.

2. Medium:

- The Project manager is a medium between his clients and his team. He must coordinate and transfer all the appropriate information from the clients to his team and report to the senior management.

3. Mentor:

- He should be there to guide his team at each step and make sure that the team has an attachment. He provides a recommendation to his team and points them in the right direction.

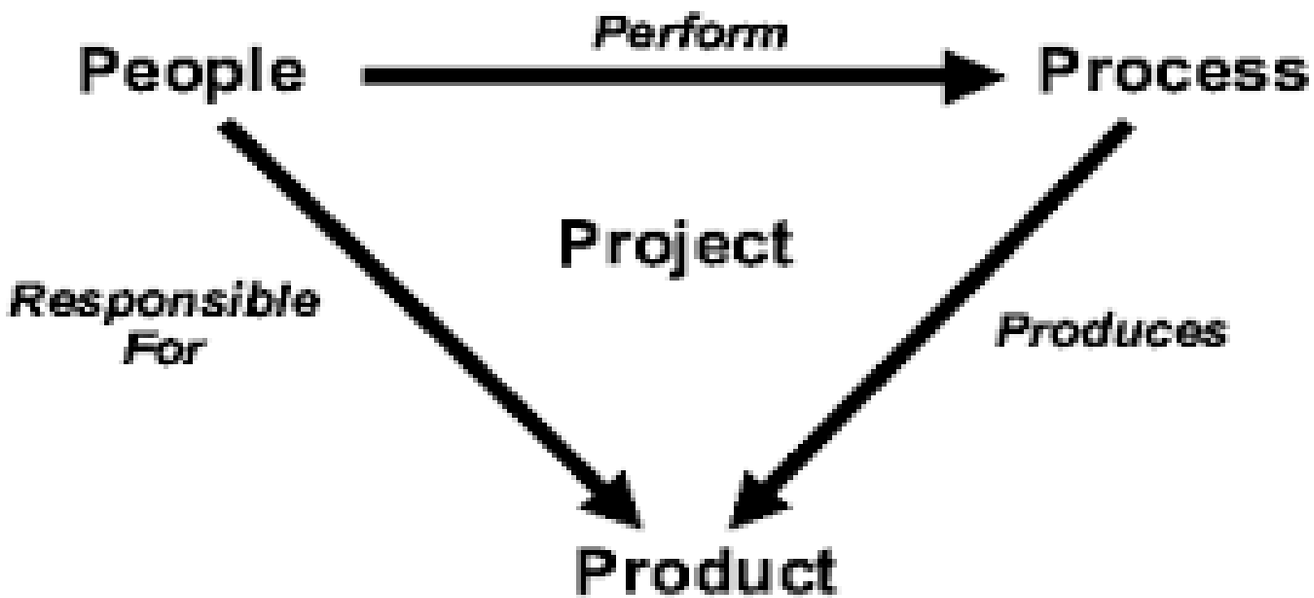
Responsibilities of a Project Manager:

- Managing risks and issues.
- Create the project team and assigns tasks to several team members.
- Activity planning and sequencing.
- Monitoring and reporting progress.
- Modifies the project plan to deal with the situation.

4 P's in Software Project Management

- For properly building a product, there's a very important concept that we all should know in software project planning while developing a product.
- There are 4 critical components in software project planning which are known as the **4P's** namely:

- Product
- Process
- People
- Project



1. People

- The most important component of a product and its successful implementation is human resources.
- In building a proper product, a well-managed team with clear-cut roles defined for each person/team will lead to the success of the product.
- We need to have a good team in order to save our time, cost, and effort. Some assigned roles in software project planning are **project manager, team leaders, stakeholders, analysts,** and other **IT professionals**.
- Managing people successfully is a tricky process which a good project manager can do.

2. Product

- As the name inferred, this is the deliverable or the result of the project.
- The project manager should clearly define the product scope to ensure a successful result, control the team members, as well technical hurdles that he or she may encounter during the building of a product.
- The product can consist of both tangible or intangible such as shifting the company to a new place or getting a new software in a company.

3. Process

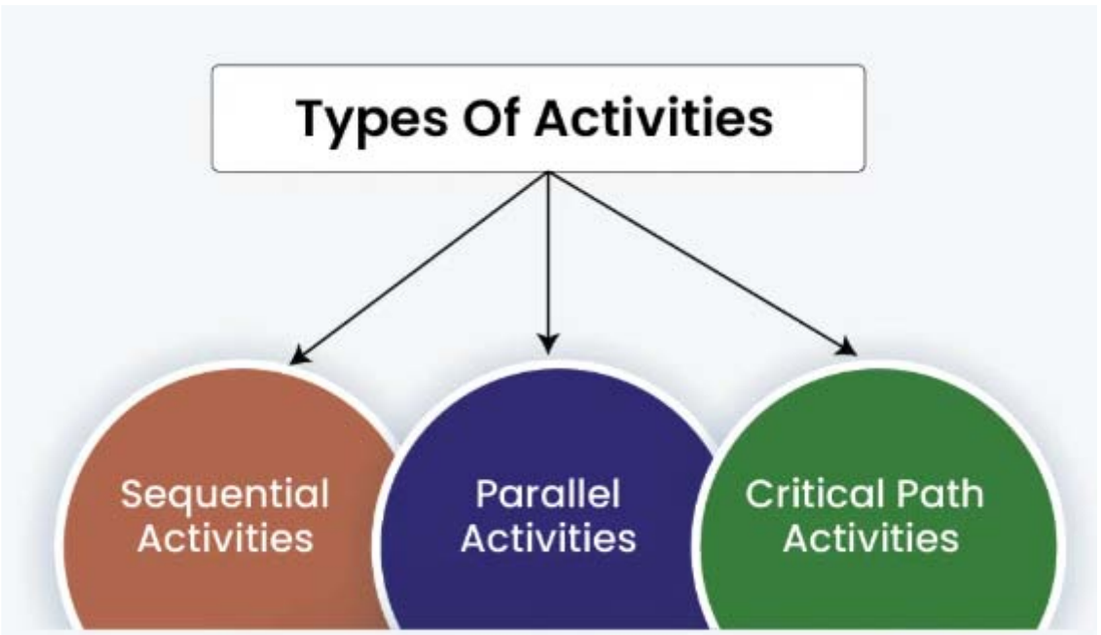
- In every planning, a clearly defined process is the key to the success of any product.
- It regulates how the team will go about its development in the respective time period.
- The Process has several steps involved like, documentation phase, implementation phase, deployment phase, and interaction phase.

4. Project

- The last and final P in software project planning is Project.
- It can also be considered as a blueprint of process.
- In this phase, the project manager plays a critical role. They are responsible to guide the team members to achieve the project's target and objectives, helping & assisting them with issues, checking on cost and budget, and making sure that the project stays on track with the given deadlines.

Activities of Project Planning

- In project management, an "**activity**" is simply a specific task or job that needs to be done for a project.
- These activities are important because they help break down the project into smaller, manageable parts, making it easier to plan and organize.
- Each activity has its own start and end dates, and it needs certain resources and people to get done.
- By identifying and defining activities, project managers can create a clear plan for the project, assign resources effectively, and keep track of progress.
- Activities are the building blocks of the project schedule, ensuring that everything gets done on time and within budget.



- **Sequential Activities:** These are tasks that must be completed in a specific order. For example, designing a product must precede manufacturing it.
- **Parallel Activities:** These are tasks that can be executed simultaneously without dependencies. For instance, while the design team works on product design, the marketing team can start creating promotional materials.
- **Critical Path Activities:** These are activities that, if delayed, would directly impact the project's overall timeline. The critical path is the longest sequence of dependent activities that determine the shortest possible duration for completing the project.

Activities

- Software Project Management consists of many activities, that includes planning of the project, deciding the scope of product, estimation of cost in different terms, scheduling of tasks, etc.

The list of activities are as follows:

- Project planning and Tracking
- Project Resource Management
- Scope Management
- Estimation Management
- Project Risk Management
- Scheduling Management
- Project Communication Management
- Configuration Management

1. **Project Planning:** It is a set of multiple processes, or we can say that it a task that performed before the construction of the product starts.
2. **Scope Management:** It describes the scope of the project. Scope management is important because it clearly defines what would do and what would not. Scope Management create the project to contain restricted and quantitative tasks, which may merely be documented and successively avoids price and time overrun.
3. **Estimation management:** This is not only about cost estimation because whenever we start to develop software, but we also figure out their size(line of code), efforts, time as well as cost.

- If we talk about the size, then Line of code depends upon user or software requirement.
- If we talk about effort, we should know about the size of the software, because based on the size we can quickly estimate how big team required to produce the software.
- If we talk about time, when size and efforts are estimated, the time required to develop the software can easily determine.

And if we talk about cost, it includes all the elements such as:

- Size of software
- Quality
- Hardware
- Communication
- Training
- Additional Software and tools
- Skilled manpower

4. **Scheduling Management:** Scheduling Management in software refers to all the activities to complete in the specified order and within time slotted to each activity. Project managers define multiple tasks and arrange them keeping various factors in mind.

For scheduling, it is compulsory -

- Find out multiple tasks and correlate them.
- Divide time into units.
- Assign the respective number of work-units for every job.
- Calculate the total time from start to finish.
- Break down the project into modules.

5. Project Resource Management

- In software Development, all the elements are referred to as resources for the project.
- It can be a human resource, productive tools, and libraries.

Resource management includes:

- Create a project team and assign responsibilities to every team member
- Developing a resource plan is derived from the project plan.
- Adjustment of resources.

6. Project Risk Management

- Risk management consists of all the activities like identification, analyzing and preparing the plan for predictable and unpredictable risk in the project.

Several points show the risks in the project:

- The Experienced team leaves the project, and the new team joins it.
- Changes in requirement.
- Change in technologies and the environment.
- Market competition.

7. Project Communication Management

- Communication is an essential factor in the success of the project.
- It is a bridge between client, organization, team members and as well as other stakeholders of the project such as hardware suppliers.
- From the planning to closure, communication plays a vital role.
- In all the phases, communication must be clear and understood. Miscommunication can create a big blunder in the project.

8. Project Configuration Management

- Configuration management is about to control the changes in software like requirements, design, and development of the product.
- The Primary goal is to increase productivity with fewer errors.

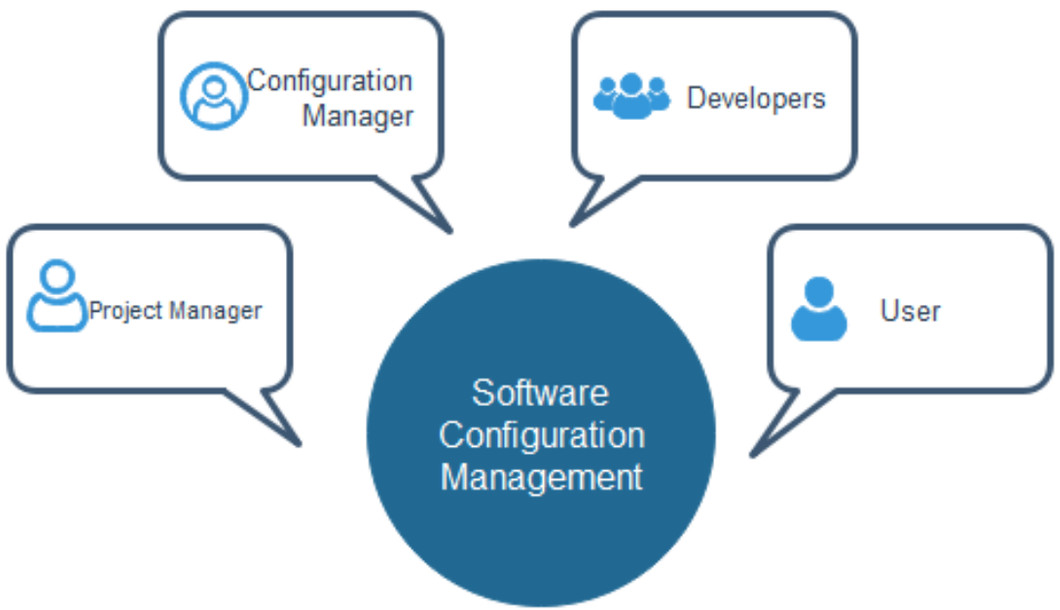
Some reasons need for configuration management:

- Several people work on software that is continually update.
- Help to build coordination among suppliers.
- Changes in requirement, budget, schedule need to accommodate.
- Software should run on multiple systems.

Tasks perform in Configuration management:

- Identification
- Baseline
- Change Control
- Configuration Status Accounting
- Configuration Audits and Reviews

People involved in Configuration Management

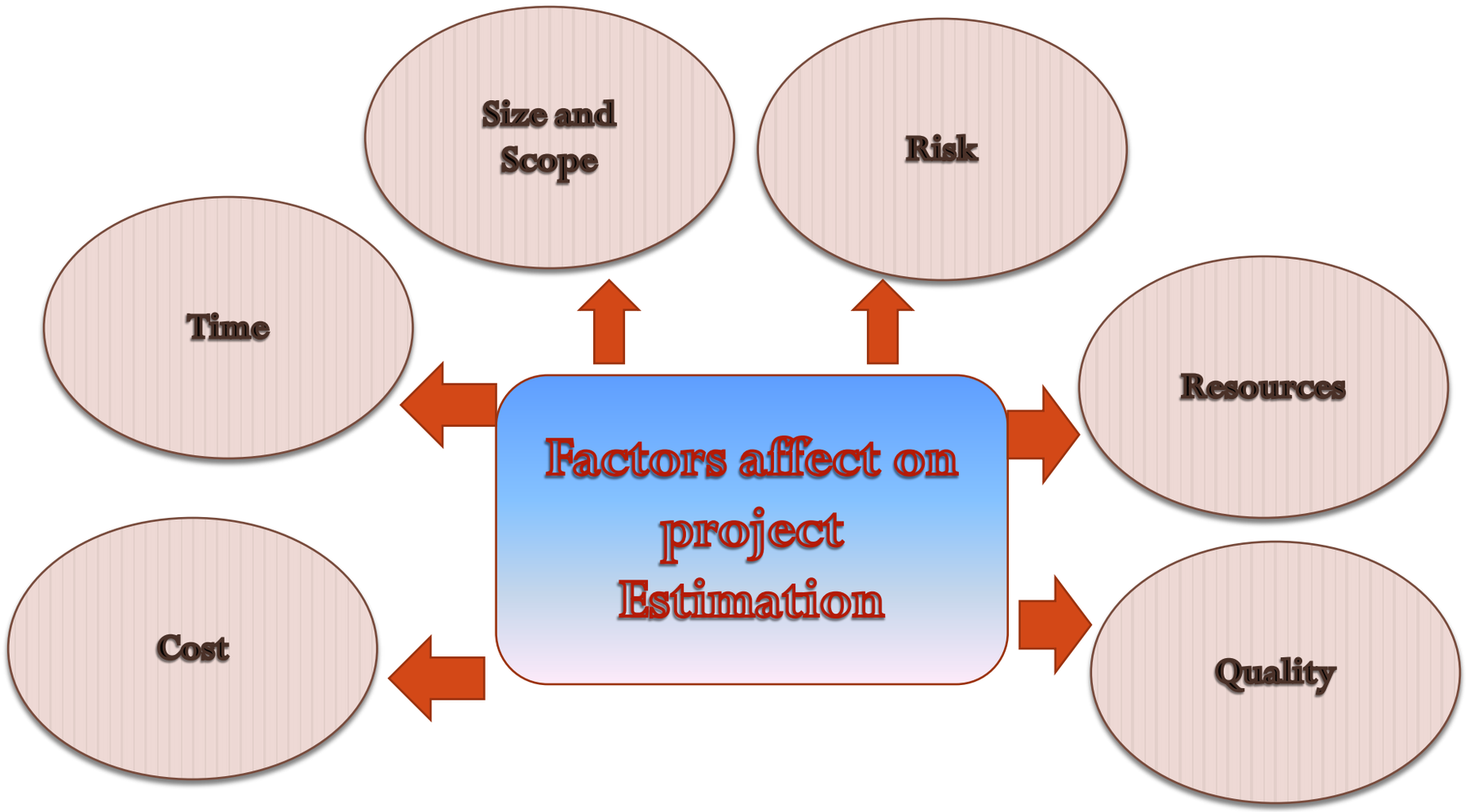


Project Estimation technique

- Software project estimation is crucial throughout the project lifecycle.
- Effective software project estimation is important activity in software development project.
- Estimation in software engineering is a procedure that predicts the time and budget that required for completing a project.
- Project Estimation requires the use of complex tools and good mathematical as well as knowledge about planning.
- One of the main reasons software programs fail is our inability to accurately estimate software size.

Factors affect on Project Estimation

1. Cost
2. Time
3. Size & Scope
4. Risk
5. Resources
6. Quality



1. Cost:

- Cost is one of the three main constraints in project management.
- If you don't have enough money to complete the project, it will fail.
- If you can accurately estimate project costs early on, you can help set client expectations and ensure you have enough money to get the work done.

2. Time

- Another of the project's three main constraints is the lack of time.
- It is critical for project planning to be able to estimate both the overall project duration and the timing of individual tasks.
- You can plan for people and resources to be available when you need them if you estimate your project schedule ahead of time.
- It also enables you to manage client expectations for key deliverables.

3. Size or Scope

- The third major project constraint is scope.
- The project scope refers to all of the tasks that must be completed in order to complete the project or deliver a product.
- You can ensure that you have the right materials and expertise on the project by estimating how much work is involved and exactly what tasks must be completed.

4. Risk

- Any unforeseen event that could positively or negatively impact your project is referred to as project risk.
- Estimating risk entails predicting what events will occur during the project's life cycle and how serious they will be.
- You can better plan for potential issues and create risk management plans if you estimate what risks could affect your project and how they will affect it.

5. Resources

- The assets you'll need to complete the project are known as project resources.
- Tools, people, materials, subcontractors, software, and other resources are all examples of resources.
- Resource management ensures that you have all of the resources you require and make the best use of them.
- It's challenging to plan how you'll manage resources without knowing what you'll need and when.
- This can result in people sitting around doing nothing or materials arriving weeks after you need them.

6. Quality

- Quality is concerned with the completion of project deliverables.
- Products that must adhere to stringent quality standards, such as environmental regulations, may require more money, time, and other resources than those with lower standards.

Project Estimation Techniques

1. Top-down estimate

- A top-down estimating technique assigns an overall time for the project and then breaks it down into discrete phases, work, and tasks — usually based on your project's work breakdown structure (WBS).
- If a client tells you the project has to be done within six months, a top-down approach allows you to take that overall timeline and estimate how much time you can take for each activity within the project and still complete it on time.

Advantages:

- Quick and simple
- Strategic Focus
- Useful for comparison
- Encourages Resource Prioritization

Disadvantages:

- Lack of Accuracy
- Potential Oversights
- Risk of over/under Allocation
- Limited Details

2. Bottom-up estimate

- A bottom-up estimate is the reverse of top-down. Using this estimation technique, you start by estimating each individual task or aspect of the project. Then you combine all those separate estimates to build up the overall project estimate.
- Since each activity is being assessed individually, this type of estimate tends to be more accurate than the top-down approach. But it also takes more time.

Advantages:

- High Accuracy
- Task-Level Insights
- Improved Risk Management
- Stakeholder Confidence

Disadvantages:

- Time-Consuming
- Complexity
- Potential for Over-Detailing
- Dependency on Input Quality

3. Expert judgment

- Expert judgment is one of the most popular estimation techniques, as it tends to be quick and easy. This technique involves relying on the experience and gut feel of experts to estimate projects.
- It's most useful when you're planning a standard project that is similar to projects your team has completed before. Expert judgment can be used for creating top-down or bottom-up estimates.

Advantages

- Quick and Simple
- Leverages Experience
- Adaptable
- Flexibility

Disadvantages

- Subjectivity
- Inconsistency
- Dependency on Experts
- Difficult to Validate

4. Comparative or analogous estimation

- Comparative estimation uses past project data combined with a top-down approach to estimate project duration.
- If the average completion time of similar projects was eight months, you'd assume the current one will take eight months. Then you can break those eight months down across tasks and activities to get your lower-level work estimates.

Advantages

- Quick and Easy
- Uses Real Data
- Early-Stage Utility
- Leverages Experience

Disadvantages

- Less Accurate
- Assumption-Driven
- Limited to Available Data
- Overlooks Unique Factors

5. Parametric model estimating

- Parametric estimating is a project estimation technique that uses statistical data, mathematical models, and historical information to estimate project costs, duration, or resource requirements. It involves identifying variables (parameters) that influence the project and applying these to calculate estimates.
- Imagine your company builds houses. Parametric modeling could take the cost of all past construction projects divided by each project's square footage to come up with an average project cost per square foot of the home. Then, you'd multiply that number by the planned square footage of the current home to create your overall project budget.

Advantages of Parametric Estimating

- High Accuracy
- Scalability
- Objective and Repeatable
- Efficiency

Disadvantages of Parametric Estimating

- Data Dependency
- Complexity
- Limited Flexibility
- Not Always Applicable

6. Three-point estimating

- Three-point estimating is a project management technique used to estimate the cost, duration, or effort required to complete a task or project with a higher level of accuracy.
- This method considers three scenarios: the most optimistic (best-case), most pessimistic (worst-case), and most likely estimates. It helps account for uncertainties and reduces the risk of underestimating or overestimating.

The three estimates used in this method are:

- **Optimistic (O):** The best-case scenario where everything goes as planned with minimal risks or issues.
- **Pessimistic (P):** The worst-case scenario considering potential delays, challenges, or risks.
- **Most Likely (M):** The most realistic scenario based on normal circumstances and historical data.

- The PERT (Program Evaluation and Review Technique) method uses three-point estimating, but it takes a weighted average of the three points, with the 'most likely' guess carrying more weight.

Advantages of Three-Point Estimating

- Accounts for Uncertainty
- Improves Accuracy
- Simple and Intuitive
- Quantifies Risk

Disadvantages of Three-Point Estimating

- Time-Consuming
- Subjectivity
- Requires Expertise
- May Not Fit All Projects

Questions

- Explain the difference between effort estimation and cost estimation.
- What are some challenges faced during project estimation?
- If a project is highly uncertain and involves new technology, which estimation technique would you recommend and why?
- A project manager has only limited historical data available. Which estimation method would you suggest they use?

COCOMO Model

- The **Constructive Cost Model** (COCOMO) is a software cost estimation model that helps predict the effort, cost, and schedule required for a software development project.
- Developed by **Barry Boehm** in 1981 and is based on the study of 63 projects, COCOMO uses a mathematical formula based on the size of the software project, typically measured in lines of code (LOC).
- The COCOMO Model is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality.

In COCOMO, projects are categorized into three types:

- Organic
- Semidetached
- Embedded

1. Organic

- A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects.
- Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

2. Semidetached

- A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff.
- Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed.
- Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.

3. Embedded

- A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the strict regulations on the operational method exist.
- **For Example:** ATM, Air Traffic control.

Mode	Project Size	Nature of Project	Innovation	Deadline of the project	Development Environment
Organic	Typically 2-50 KLOC	Small size project, experienced developers in the familiar environment. For example, Payroll ,Inventory projects etc.	Little	Not tight	Familiar & In house
Semi detached	Typically 50-300 KLOC	Medium size project , Medium size team, Average previous experience on similar project. For example Utility systems like compilers, database system, editors etc.	Medium	Medium	Medium
Embedded	Typically over 300 KLOC	Large project, Real time systems, Complex interfaces, very little experience. For example: ATMs , Air traffic control etc.	Significant	Tight	Complex Hardware/Customer Interface required

Person Month(PM)

- Person month is a measurement unit for effort in software engineering. 1 person month means effort put by a person in one month.
- The effort estimation is expressed in units of person-month (PM).
- An effort of 100PM does not imply that 100 persons should work for 1 month nor does it imply that 1 person should be employed for 100 months, but it denotes the area under the person-month curve.
- It is the area under the person-month plot.
- **Example:** 3 Developer work for 4 month , it means 12 PM effort is applied.

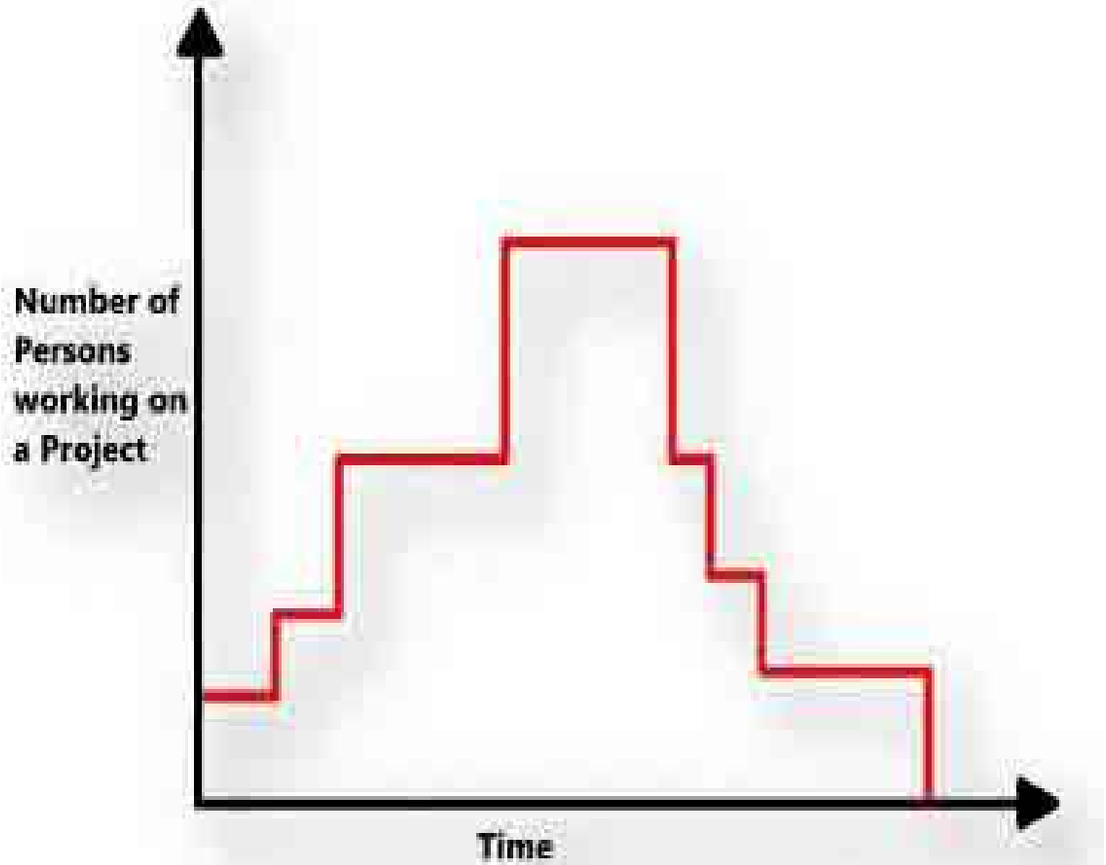
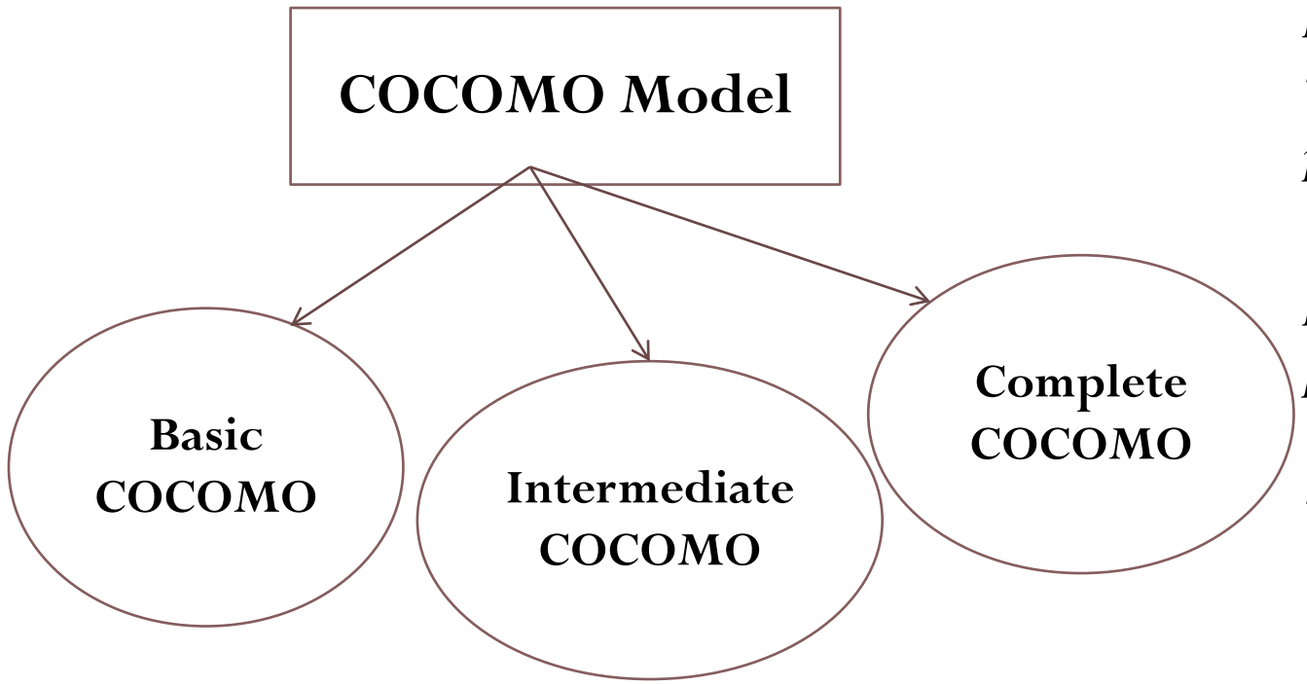


Fig. Person Month Curve

- According to Boehm, software cost estimation should be done through three stages:



1. Basic COCOMO Model

- Basic COCOMO Model computes software development effort, time, and cost as a function of program size. Program size is expressed in estimated thousands of source line of code (SLOC, KLOC).

- KLOC is the estimated number of delivered line (expressed in thousands) of code for project, estimated size of software product.

$E = a*(KLOC)^b\ PM$

$Tdev = c*(E)^d$

Person required = Effort / Time

Where,

E is effort applied in Person-Months

KLOC is the estimated size of the software product indicate in Kilo Lines of Code

Tdev is the development time in months

Effort is the total effort required to develop the software product, expressed in **person months (PMs)**.

Example of Basic COCOMO Model

- Suppose that a Basic project was estimated to be 400 KLOC (kilo lines of code). Calculate effort and time for each of the three modes of development. All the constants value provided in the table:

Solution

From the above table we take the value of constant a,b,c and d.

For organic mode,

effort = $2.4 \times (400)^{1.05} \approx 1295$ person-month.

dev. time = $2.5 \times (1295)^{0.38} \approx 38$ months.

For semi-detach mode,

effort = $3 \times (400)^{1.12} \approx 2462$ person-month.

dev. time = $2.5 \times (2462)^{0.35} \approx 38$ months.

For Embedded mode,

effort = $3.6 \times (400)^{1.20} \approx 4772$ person-month.

dev. time = $2.5 \times (4772)^{0.32} \approx 38$ months.

Category	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Example: Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three model i.e., organic, semi-detached & embedded.

Solution: The basic COCOMO equation takes the form:

$Effort = a*(KLOC)^{b}$ PM

$Tdev = c*(efforts)^{d}$ Months

Estimated Size of project= 400 KLOC

(i)Organic Mode

$E = 2.4 * (400)^{1.05} = 1295.31$ PM

$D = 2.5 * (1295.31)^{0.38} = 38.07$ PM

(ii)Semidetached Mode

$E = 3.0 * (400)1.12 = 2462.79$ PM

$D = 2.5 * (2462.79)0.35 = 38.45$ PM

(iii) Embedded Mode

$E = 3.6 * (400)1.20 = 4772.81$ PM

$D = 2.5 * (4772.8)0.32 = 38$ PM

Example: A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

Solution:

- The semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence

$E=3.0(200)^{1.12}=1133.12PM$

$D=2.5(1133.12)^{0.35}=29.3PM$

$$\text{Average Staff Size (SS)} = \frac{E}{D} \text{ Persons}$$

$$= \frac{1133.12}{29.3} = 38.67 \text{ Persons}$$

$$\text{Productivity} = \frac{KLOC}{E} = \frac{200}{1133.12} = 0.1765 \text{ KLOC/PM}$$

P = 176 LOC/PM

Merits

- Basic COCOMO is good for quick , rough and early estimate of software costs.

Demerits

- It does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques and so on.
- The accuracy of this model is limited because it does not consider certain factors for cost estimation of software.

2. Intermediate COCOMO Model

- The basic COCOMO model considers that the effort is only a function of the number of lines of code and some constants calculated according to the various software systems.
- The intermediate COCOMO model recognizes these facts and refines the initial estimates obtained through the basic COCOMO model by using a set of 15 cost drivers based on various attributes of software engineering.

Classification of Cost Drivers and their Attributes:

The cost drivers are divided into four categories

- **Product attributes:**
  - Required software reliability extent
  - Size of the application database
  - The complexity of the product
- **Hardware attributes**
  - Run-time performance constraints
  - Memory constraints
  - The volatility of the virtual machine environment
  - Required turnabout time
- **Personal attributes**
  - Analyst capability
  - Software engineering capability
  - Application experience
  - Virtual machine experience
  - Programming language experience
- **Project attributes**
  - Use of software tools
  - Application of software engineering methods
  - Required development schedule

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Software project	a_i	b_i
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

- Consider a project having 30000 lines of code which in an embedded software with critical area hence reliability is high. Calculate effort, duration and size of staff.

Solution:

Here given, LOC=30000 =30KLOC

As reliability is high EAF=1.15 (Product attribute)

Since project is embedded,

a_i =2.8 and b_i =1.20

$E= a_i (KLOC)^{bi}$ x (EAF) → *effort adjustment factor*

$E=2.8 (30)^{1.20}$ x 1.15

=191 PM

$D=c(E)^d$

$=2.5(191)^{0.32}$

=13 Month approx

N=E/D

=191/13

N=15 person approx

Note: The value of c and d use from Basic COCOMO model.

❖ Consider a project having 20000 lines of code which in an semi-detached software with critical area hence reliability is very high. Calculate effort, duration and size of staff.

Solution:

Here given, LOC=20000 =20KLOC

As reliability is very high EAF=1.40 (Product attribute)

Since project is semi-deatched,

a_i =3.0 and b_i =1.12

E= a_i (KLOC)^{bi} x (EAF) → *effort adjustment factor*

E=3.0 $(20)^{1.12}$ x 1.40

=120 PM

D=c$(E)^d$

=2.5$(120)^{0.35}$

=13 Month approx

N=E/D

=120/13

N=9 person approx

Merits:

- This model can be applied to almost to entire software product for easy and rough cost estimation during early stage.
- It can be applied at the software product component level for obtaining more accurate cost estimation.

Demerits:

- The effort multipliers are not dependent on phases.
- A product with many components is difficult to estimate.

3. Detailed COCOMO Model

- Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost drivers effect on each method of the software engineering process.
- The detailed model uses various effort multipliers for each cost driver property.
- The complete COCOMO model considers the differences in characteristics of all the subsystems and estimates the effort and development time as sum of the estimates for the individual sub system.
- In detailed COCOMO, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.
- Cost of each sub-system is estimated separately.
- Costs of the sub-systems are added to obtain total cost.
- Reduces the margin of error in the final estimate.

The Six phases of detailed COCOMO are:

- Planning and requirements
- System structure
- Complete structures
- Module code and test
- Integration and test
- Cost Constructive model

Example:

A management information system (MIS) for an organization having offices at several places across the country:

- Database part (semi-detached)
- Graphical user Interface (GUI) part (organic)
- Communication part (embedded)

Costs of the components are estimated separately :

- Summed up to give the overall cost of the system.

What is Risk

- "Tomorrow problems are today's risk." Hence, a clear definition of a "risk" is a problem that could cause some loss or threaten the progress of the project, but which has not happened yet.
- These potential issues might harm cost, schedule or technical success of the project and the quality of our software device, or project team morale.
- Risk Management is the system of identifying addressing and eliminating these problems before they can damage the project.
- For example, staff storage, because we have not been able to select people with the right technical skills is a current problem, but the threat of our technical persons being hired away by the competition is a risk.

Risk Management

- A software project can be concerned with a large variety of risks. In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes.
- The project manager can then check which risks from each class are relevant to the project.

There are five main classifications of risks which can affect a software project:

- Schedule risks
- Budget risks
- Operational risks
- Technical risks
- Programmatic Risks

1. Schedule Risk :

- Schedule related risks refers to time related risks or project delivery related planning risks. The wrong schedule affects the project development and delivery.
- These risks are mainly indicates to running behind time as a result project development doesn't progress timely and it directly impacts to delivery of project.
- If schedule risks are not managed properly it gives rise to project failure and at last it affect to organization/company economy very badly.
- Some reasons for Schedule risks:
  - Time is not estimated perfectly
  - Improper resource allocation
  - Tracking of resources like system, skill, staff etc
  - Frequent project scope expansion
  - Failure in function identification and its' completion

2. **Budget Risk:** Budget related risks refers to the monetary risks mainly it occurs due to budget overruns. Always the financial aspect for the project should be managed as per decided but if financial aspect of project mismanaged then there budget concerns will arise by giving rise to budget risks. So proper finance distribution and management are required for the success of project otherwise it may lead to project failure. Some reasons for Budget risks –
   - Wrong/Improper budget estimation
   - Unexpected Project Scope expansion
   - Mismanagement in budget handling
   - Cost overruns
   - Improper tracking of Budget

3. Operational Risks:

- Operational risk refers to the procedural risks means these are the risks which happen in day-to-day operational activities during project development due to improper process implementation.

Some reasons for Operational risks:

- Insufficient resources
- Conflict between tasks and employees
- Improper management of tasks
- No proper planning about project
- Less number of skilled people
- Lack of communication and cooperation
- Lack of clarity in roles and responsibilities
- Insufficient training

4. Technical Risks

- Technical risks refers to the functional risk or performance risk which means this technical risk mainly associated with functionality of product or performance part of the software product.
- Some reasons for Technical risks:
  - Frequent changes in requirement
  - Less use of future technologies
  - Less number of skilled employee
  - High complexity in implementation
  - Improper integration of modules

5. Programmatic Risks

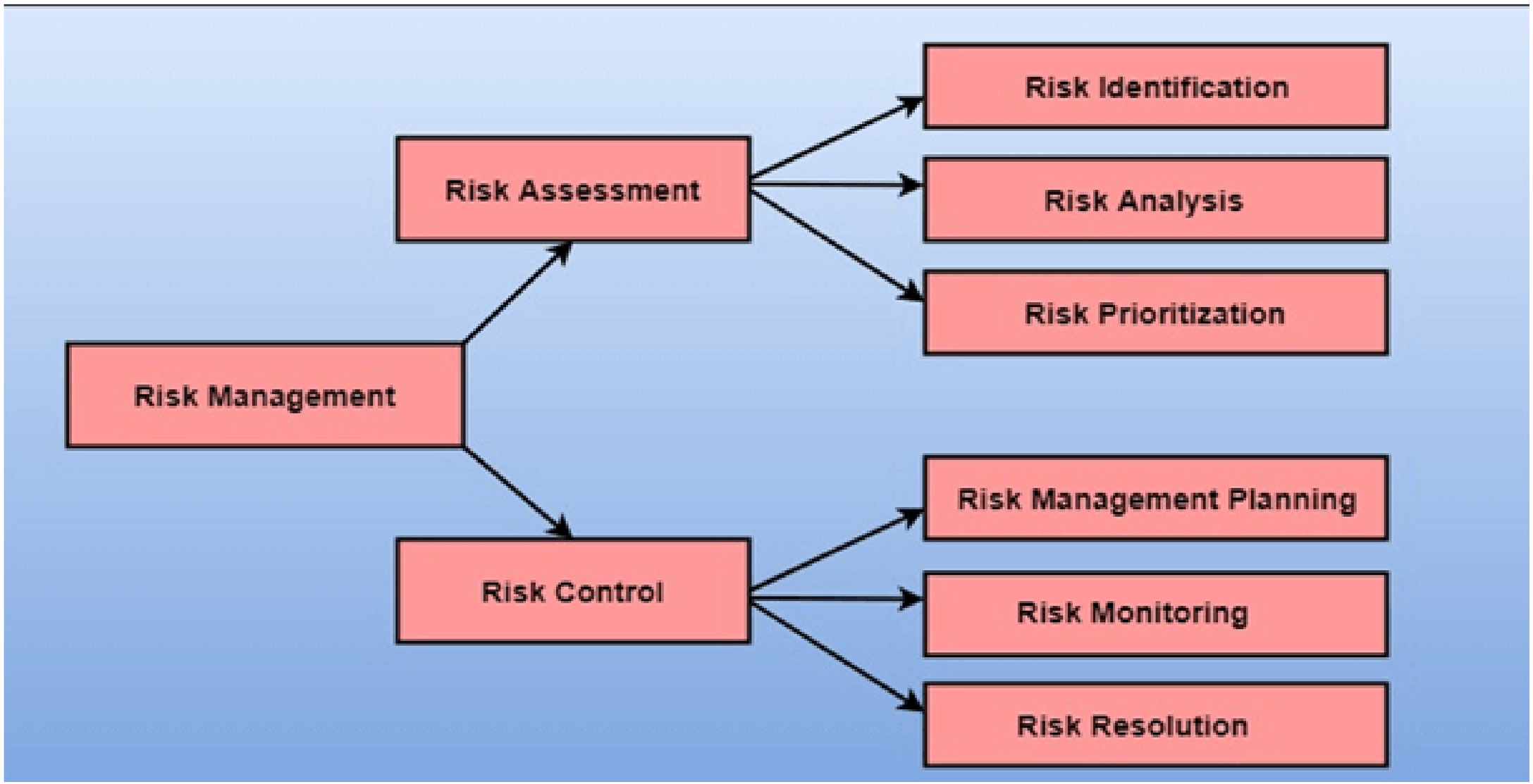
- Programmatic risks refers to the external risk or other unavoidable risks.
- These are the external risks which are unavoidable in nature.
- These risks come from outside and it is out of control of programs.
- Some reasons for Programmatic risks:
  - Rapid development of market
  - Running out of fund / Limited fund for project development
  - Changes in Government rules/policy
  - Loss of contracts due to any reason

More risks associated with software development

- **Communication Risks:** Misunderstandings, mistakes and a general sense of confusion can result from inadequate or absent communication.
- **Security Risks:** Vulnerabilities that might compromise the privacy, reliability or accessibility of the set are known as security risks and they have become common in a time.
- **Quality Risks:** The risk associated with quality is the potential for a product to be delivered that does not meet end user satisfaction or required criteria.
- **Risks associated with Law and Compliance:** Rules and laws are often overlooked when it comes to project development. Ignoring them may result in penalties, legal issues or just a lot of difficulties.
- **Cost Risks:** Unexpected costs, changes in the project scope or excess funds may completely halt your financial plan.
- **Market Risks:** The effectiveness of your programme in the market may be compromised by evolving technology trends, new competitors or shifting the customer wants.

Risk Management Activities

Risk management consists of two main activities



1. Risk Assessment

- The objective of risk assessment is to division the risks in the condition of their loss, causing potential.
- For risk assessment, first, every risk should be rated in two methods:
  - The possibility of a risk coming true (r).
  - The consequence of the issues relates to that risk (s).

A. Risk Identification

- The project organizer needs to anticipate the risk in the project as early as possible so that the impact of risk can be reduced by making effective risk management planning.

There are different types of risks which can affect a software project:

- **Technology risks:** Risks that assume from the software or hardware technologies that are used to develop the system.
- **People risks:** Risks that are connected with the person in the development team.
- **Organizational risks:** Risks that assume from the organizational environment where the software is being developed.
- **Tools risks:** Risks that assume from the software tools and other support software used to create the system.

- **Requirement risks:** Risks that assume from the changes to the customer requirement and the process of managing the requirements change.
- **Estimation risks:** Risks that assume from the management estimates of the resources required to build the system.

B. Risk Analysis

- During the risk analysis process, you have to consider every identified risk and make a perception of the probability and seriousness of that risk.
- It is not possible to make an exact, the numerical estimate of the probability and seriousness of each risk. Instead, you should authorize the risk to one of several bands:
- The probability of the risk might be determined as very low (0-10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (+75%).

C. Risk prioritization

- The risk priority can be identified using the formula below
- Based on these two methods, the priority of each risk can be estimated:

$$p = r * s$$

- Where p is the priority with which the risk must be controlled, r is the probability of the risk becoming true, and s is the severity of loss caused due to the risk becoming true.

2. Risk Control

- It is the process of managing risks to achieve desired outcomes.
- After all, the identified risks of a plan are determined; the project must be made to include the most harmful and the most likely risks.
- Different risks need different containment methods. In fact, most risks need ingenuity on the part of the project manager in tackling the risk.

There are three main methods to plan for risk management:

- **Avoid the risk:** This may take several ways such as discussing with the client to change the requirements to decrease the scope of the work, giving incentives to the engineers to avoid the risk of human resources turnover, etc.
- **Transfer the risk:** This method involves getting the risky element developed by a third party, buying insurance cover, etc.
- **Risk reduction:** This means planning method to include the loss due to risk. For instance, if there is a risk that some key personnel might leave, new recruitment can be planned.

A. Risk planning

- The risk planning method considers each of the key risks that have been identified and develop ways to maintain these risks.
- For each of the risks, you have to think of the behavior that you may take to minimize the disruption to the plan if the issue identified in the risk occurs.
- You also should think about data that you might need to collect while monitoring the plan so that issues can be anticipated.
- Again, there is no easy process that can be followed for contingency planning. It rely on the judgment and experience of the project manager.

B. Risk Monitoring

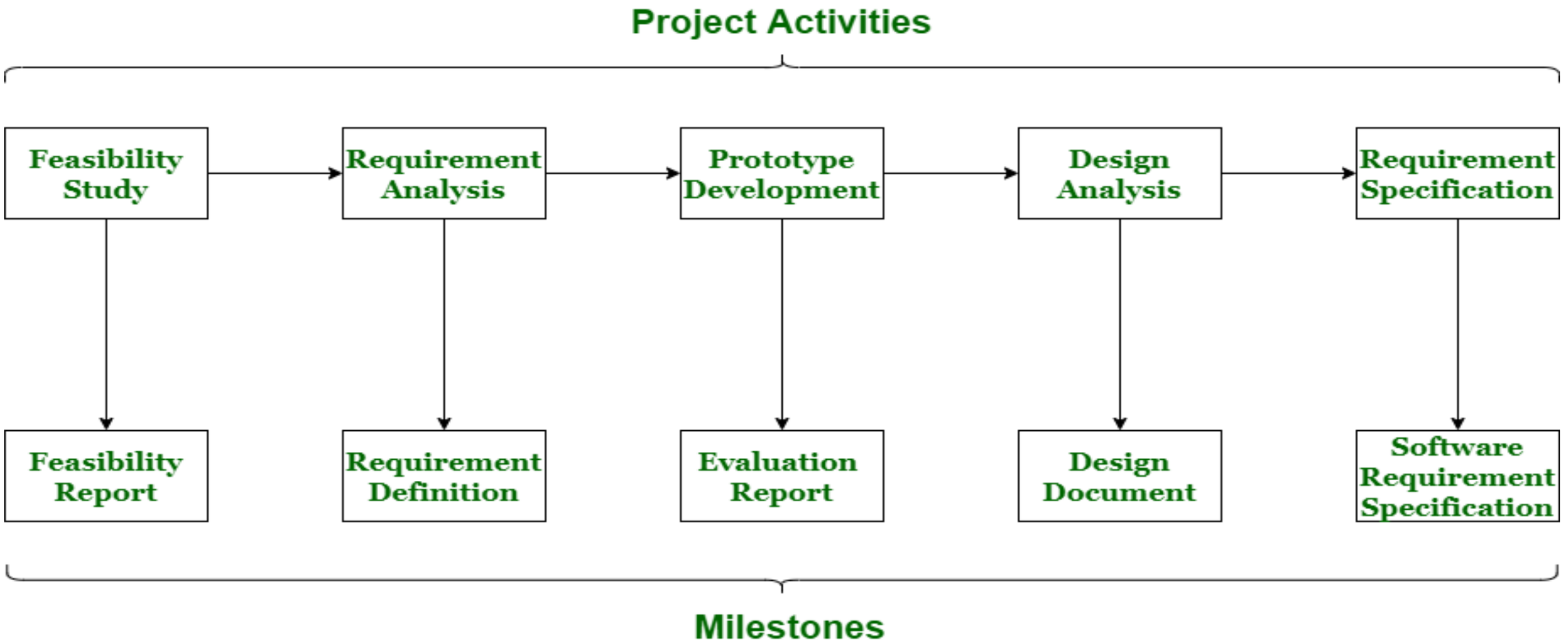
- Risk monitoring is the method king that your assumption about the product, process, and business risks has not changed.

C. Risk resolution

- It involves eliminating the overall risk or finding solutions.
- This method includes techniques such as design to cost approach, simulating the prototype, benchmarking, etc.

Milestone :

- When project begins then it is expected that project related activities must be initiated.
- In project planning, series of milestones must be established.
- Milestone can be defined as recognizable endpoint of software project activity.
- At each milestone, report must be generated.
- Milestone is distinct and logical stage of the project.
- It is used as signal post for project start and end date, need for external review or input and for checking budget, submission of the deliverable, etc.
- It simply represents clear sequence of events that are incrementally developed or build until project gets successfully completed.
- It is generally referred to as task with zero-time duration because they are used to symbolize an achievement or point of time in project.
- It helps in signifying change or stage in development.



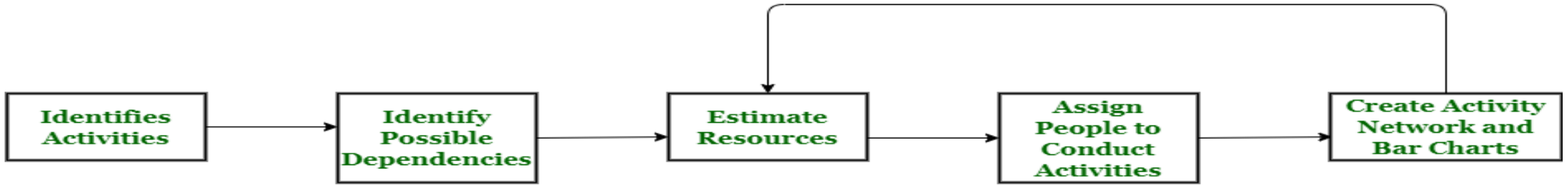
Various Project Activities and Milestones

Deliverable :

- It simply means result or software product, designed document, or asset of project plan that can be submitted to customers, clients, or end-users.
- A deliverable should be completed in all aspects.
- It is an element of output within scope of project or processes in the project.
- The deliverables have a due date, are real and touchable, and measurable.
- The deliverable is simply given to client or customer and satisfies milestone or due date that is often created and produced during project planning.
- Deliverables are generally milestones but it not necessary that milestone is deliverable.

Project Scheduling

- **Project schedule** simply means a mechanism that is used to communicate and know about that tasks are needed and has to be done or performed and which organizational resources will be given or allocated to these tasks and in what time duration or time frame work is needed to be performed.
- Effective project scheduling leads to success of project, reduced cost, and increased customer satisfaction.
- Scheduling in project management means to list out activities, deliverables, and milestones within a project that are delivered.
- It contains more notes than your average weekly planner notes. The most common and important form of project schedule is Gantt chart.



Project Scheduling Process

Process

- The manager needs to estimate time and resources of project while scheduling project.
- All activities in project must be arranged in a coherent sequence that means activities should be arranged in a logical and well-organized manner for easy to understand.
- Initial estimates of project can be made optimistically which means estimates can be made when all favorable things will happen and no threats or problems take place.
- The total work is separated or divided into various small activities or tasks during project schedule. Then, Project manager will decide time required for each activity or task to get completed.
- Even some activities are conducted and performed in parallel for efficient performance. The project manager should be aware of fact that each stage of project is not problem-free.

To schedule the project plan, a software project manager wants to do the following:

- Identify all the functions required to complete the project.
- Break down large functions into small activities.
- Determine the dependency among various activities.
- Establish the most likely size for the time duration required to complete the activities.
- Allocate resources to activities.
- Plan the beginning and ending dates for different activities.
- Determine the critical path. A critical way is the group of activities that decide the duration of the project.

Advantages of Project Scheduling

There are several advantages provided by project schedule in our project management:

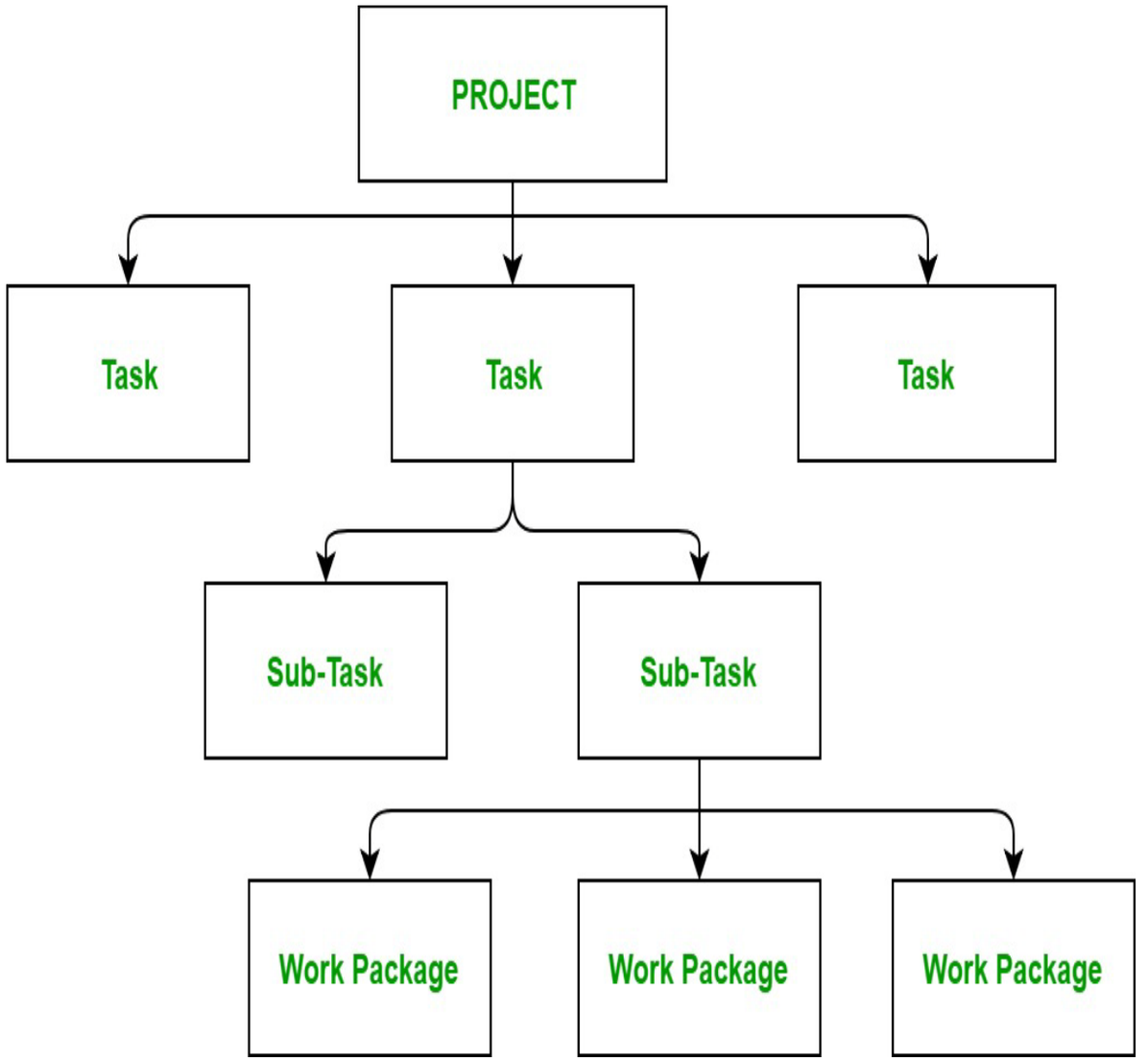
- It simply ensures that everyone remains on same page as far as tasks get completed, dependencies, and deadlines.
- It helps in identifying issues early and concerns such as lack or unavailability of resources.
- It also helps to identify relationships and to monitor process.
- It provides effective budget management and risk mitigation.

Project Scheduling Techniques

1. Work Breakdown Structure(WBS)
2. Activity Chart
3. Gantt Charts
4. PERT
5. Critical Path Method (CPM)

Work Breakdown Structure

- Work breakdown structure(WBS) in project management and system engineering, is a deliverable-oriented decomposition of a project into smaller components.
- A Work Breakdown Structure (WBS) is a fundamental project management tool that visually represents the hierarchical decomposition of a project into smaller, more manageable components or tasks.
- It's like a roadmap that guides the project team from the overall project objectives to the specific activities required to achieve them.



Steps Work Breakdown Structure:

Step 1: Identify the major activities of the project.

Step 2: Identify the sub-activities of the major activities.

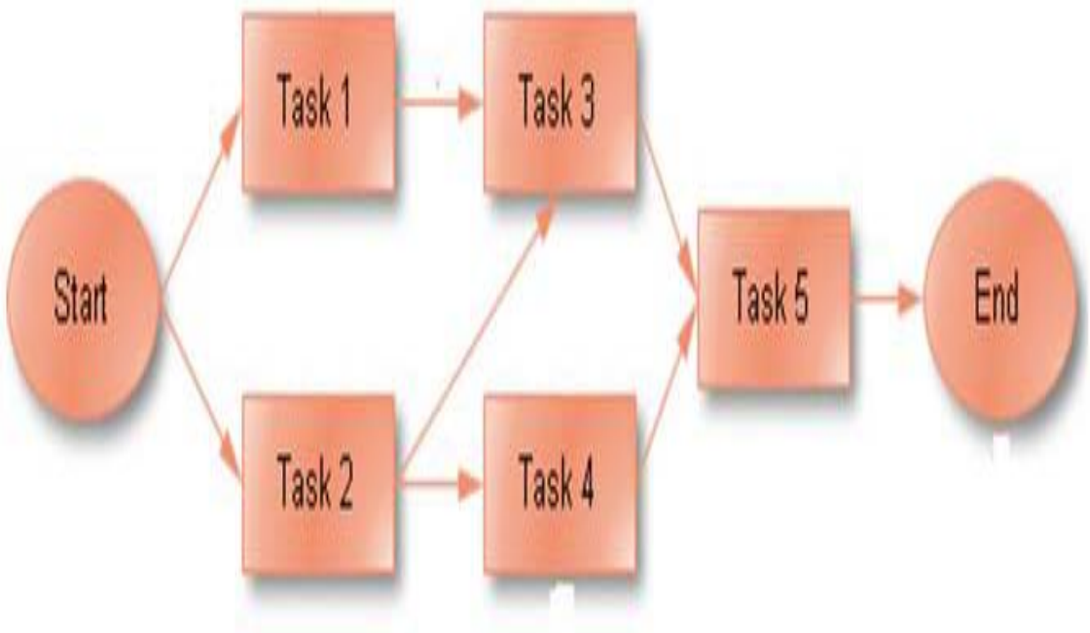
Step 3: Repeat till undividable, simple, and independent activities are created.

Construction of Work Breakdown Structure

- Firstly, the project managers and top level management identifies the main deliverables of the project.
- After this important step, these main deliverables are broke down into smaller higher-level tasks and this complete process is done recursively to produce much smaller independent tasks.
- It depends on the project manager and team that upto which level of detail they want to break down their project.
- Generally the lowest level tasks are the most simplest and independent tasks and takes less than two weeks worth of work.
- Hence, there is no rule for upto which level we may build the work breakdown structure of the project as it totally depends upon the type of project we are working on and the management of the company.
- The efficiency and success of the whole project majorly depends on the quality of the Work Breakdown Structure of the project and hence, it implies its importance.

Activity Chart

- When it comes to a project , the entire project is divided into may interdependent tasks. In this set of tasks, the sequence or the order of the tasks is quite important.
- If the sequence is wrong, the end result of the project might not be what the management excepted.
- Some tasks in the projects can safely be performed parallel to other tasks. In a project activity diagram, the sequence of the tasks is simply illustrated.
- There are many tools that can be used for drawing project activity diagrams. Microsoft project is one of the most popular software for this type of work.



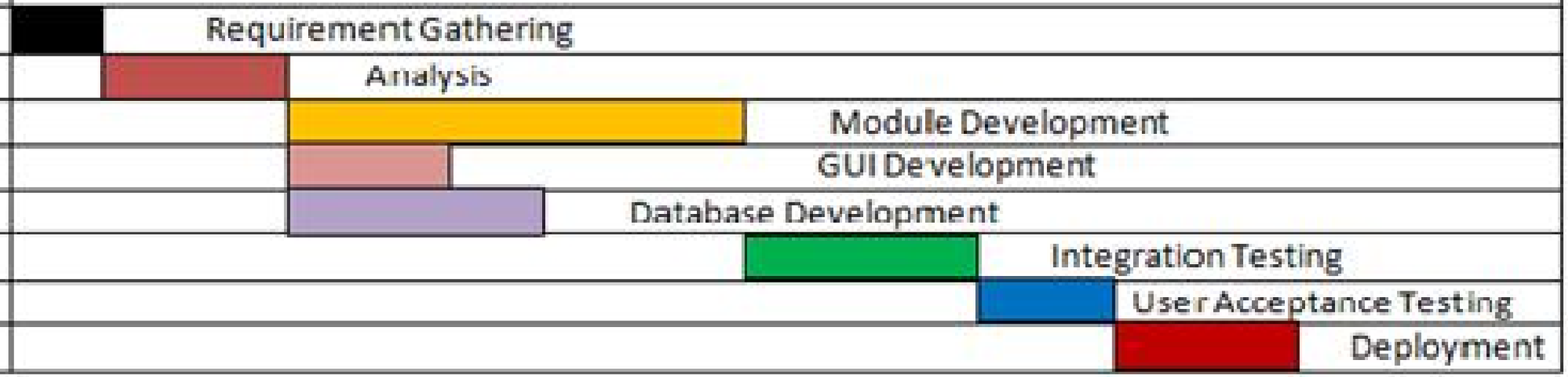
Gantt Chart:

- Gantt chart is one of the most commonly used methods for project planning and scheduling.
- This chart typically depicts activities as horizontal bars whose length depends on the duration needed for completing the activities.
- In order to show how much of these activities completed, the horizontal bars can be overprinted during the execution of project.

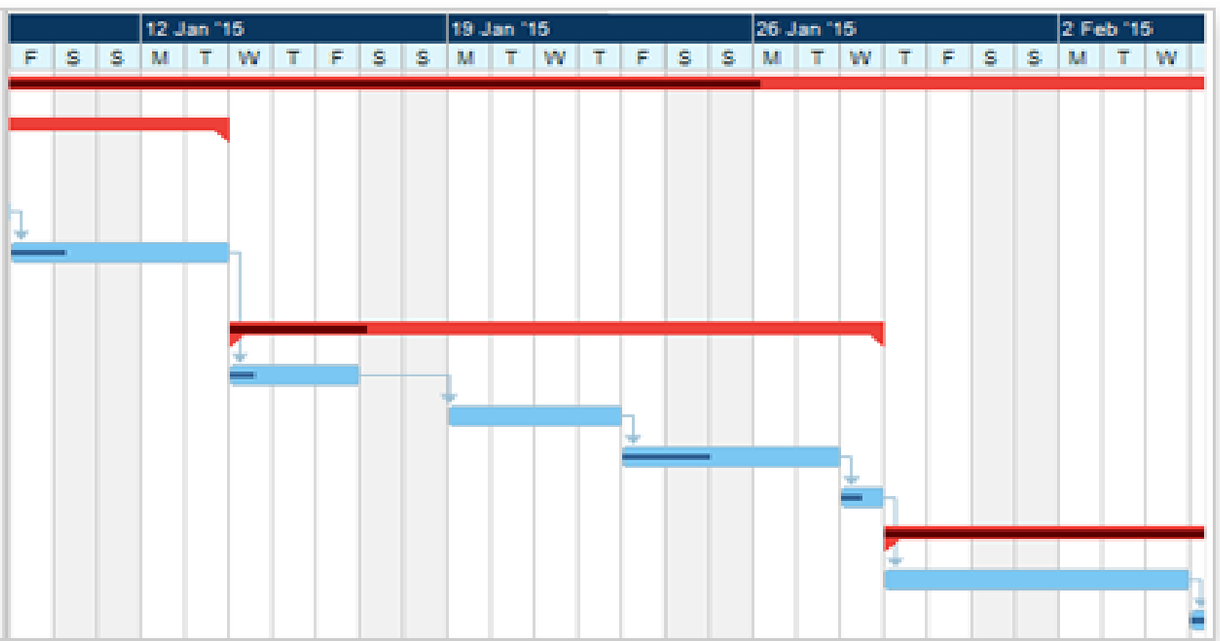
A Gantt chart allows you to simplify complex projects into an easy-to-follow plan that includes:

- How a project breaks down into tasks
- When each task will begin and end
- How long each task will take
- Who's assigned to each task
- How tasks relate to and depend on each other
- When important meetings, approvals, or deadlines need to happen
- How work is progressing in a project
- The full project schedule from start to finish

Task	Duration (days)
A	10
B	30
C	90
D	25
E	40
F	35
G	20
H	25

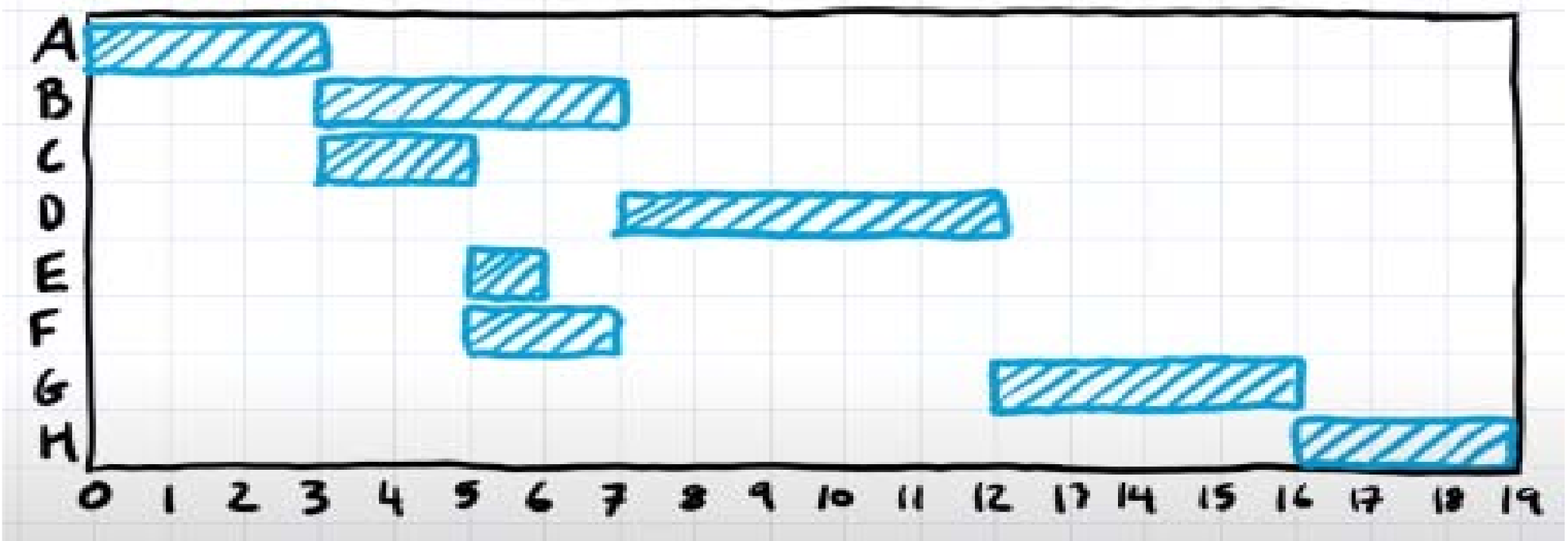


	Tas...	Task Name	Duration	Start	End
1		Business Plan	48 days?	1/1/2015	3/9/2015
2		⊟ Phase 1 - Str...	9 days	1/1/2015	1/13/201!
3		Define the...	2 days	1/1/2015	1/2/2015
4		Revenue P...	4 days	1/5/2015	1/8/2015
5		Evaluate B...	3 days	1/9/2015	1/13/201!
6		Helpful Links	0 days	1/1/2015	1/1/2015
7		⊟ Phase 2 - De...	11 days?	1/14/2015	1/28/201!
8		Define the...	3 days	1/14/2015	1/16/201!
9		Identify Ne...	4 days	1/19/2015	1/22/201!
10		Evaluate P...	3 days	1/23/2015	1/27/201!
11		Confirm de...	1 day?	1/28/2015	1/28/201!
12		⊟ Phase 3 - Pla...	15 days	1/29/2015	2/18/201!
13		Develop D...	5 days	1/29/2015	2/4/2015
14		Describe th...	2 days	2/5/2015	2/6/2015



Activity	Predecessor	Duration
A	—	3
B	A	4
C	A	2
D	B	5
E	C	1
F	C	2
G	D, E	4
H	F, G	3

 # Create Gantt chart using following information.



Program Evaluation and Review Technique (PERT)

- A PERT chart, also known as a PERT diagram, is a tool used to schedule, organize, and map out tasks within a project.
- It provides a visual representation of a project's timeline and breaks down individual tasks.
- In a project scenario, usually an ‘activity’ is a task to be performed and a ‘milestone’ is that marks the completion of a single or multiple activities.
- In PERT, the activities are represented on the arcs between nodes and milestones on the nodes.
- Generally the milestones will be numbered so that the starting node of an activity will be having a lower number than the ending node.

- The initial step in PERT is to identify the activities and milestones.
- The activity sequence in which the different activities must be performed is determined.
- A network diagram will be created using the activity sequence information.
- It will depict the sequence of serial and parallel activities. As next step, the time for activity completion is estimated.

Usually the model includes three time estimates for each activity.

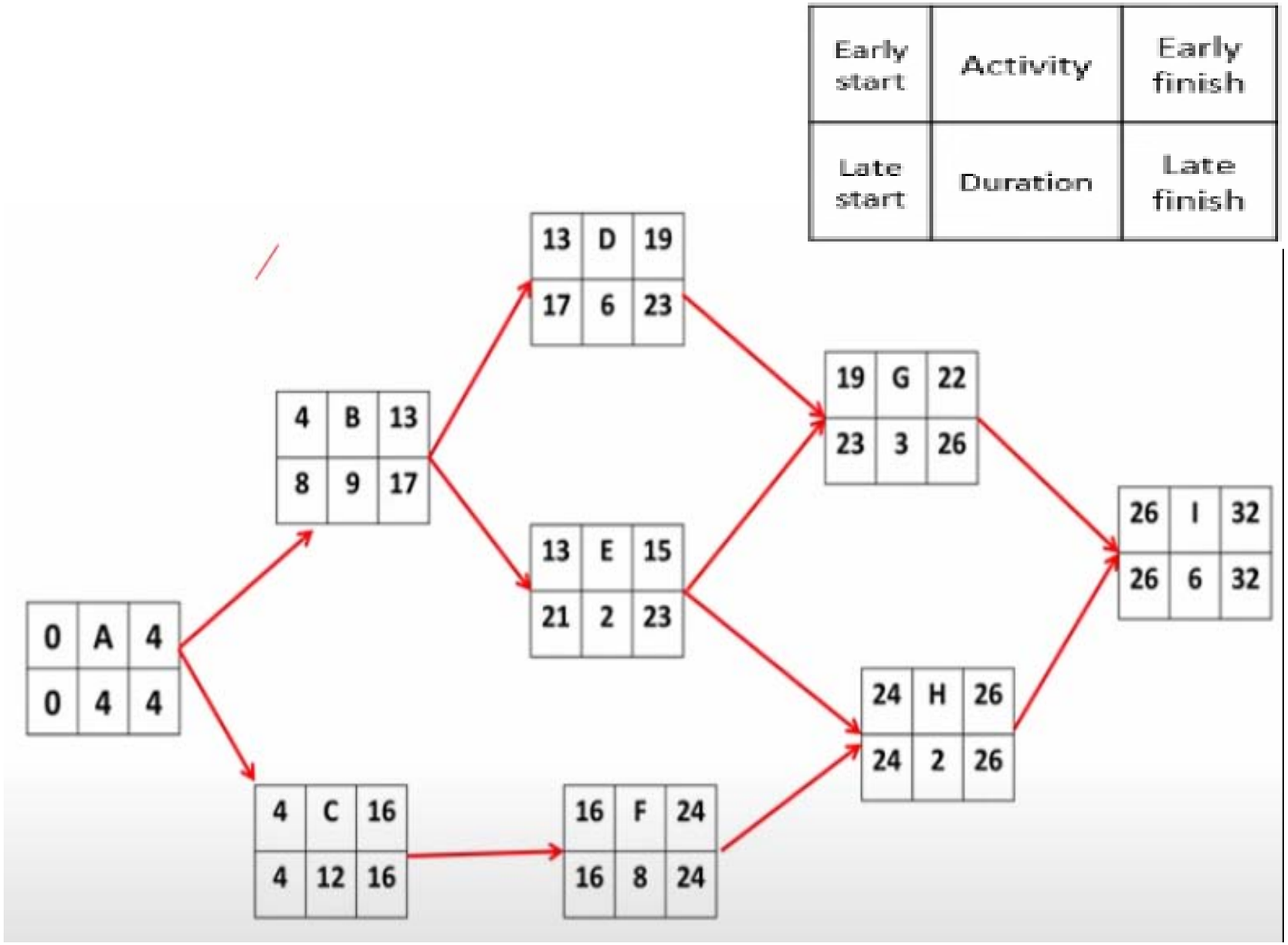
Optimistic time (OT): The shortest time in which the activity could be completed.

Most likely time (MT): The time which will be most probably required for the activity to complete.

Pessimistic time (PT): The longest time for completion.

- Using these values, the expected time for each activity can be calculated as:
- Expected time = (OT + 4 x MT+ PT) / 6

Activity	Predecessor	Timing			M. time
		A	M	B	
A		1	5	3	4
B	A	2	8	20	9
C	A	8	12	16	12
D	B	4	6	8	6
E	B	1	2	3	2
F	C	6	8	10	8
G	E,D	2	3	4	3
H	E,F	2	2	2	2
I	H,G	6	6	6	6



Critical Path Method(CPM)

- Critical Path Method (CPM) is a method used in project planning, generally for project scheduling for the on-time completion of the project.
- It helps in the determination of the earliest time by which the whole project can be completed.
- The Critical Path Method (CPM) is the most commonly used technique for construction scheduling.
- The critical path refers to the longest stretch of the activities.
- The duration of the critical path method represents the time required to complete a construction project.

The critical path method helps to decide the following things:

- The duration to complete an activity
- The dependencies between the activities
- The tasks required to complete the project

How to find the critical path in a project:

Step 1: Identify all tasks required to complete the project

Step 2: Determine the sequence of tasks

Step 3: Estimate the duration of each task

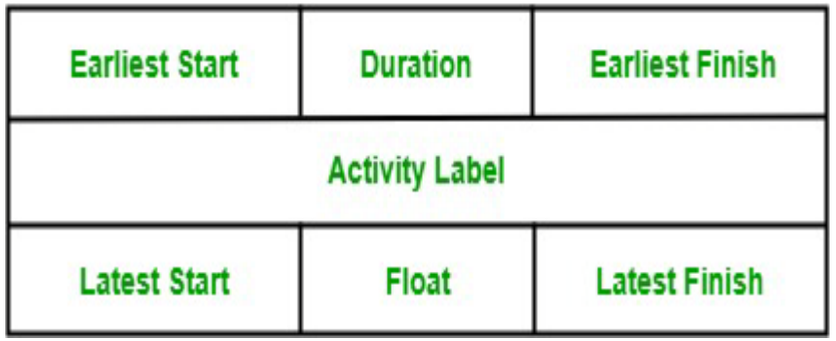
Step 4: Draw a network diagram

Step 5: Identify the critical path

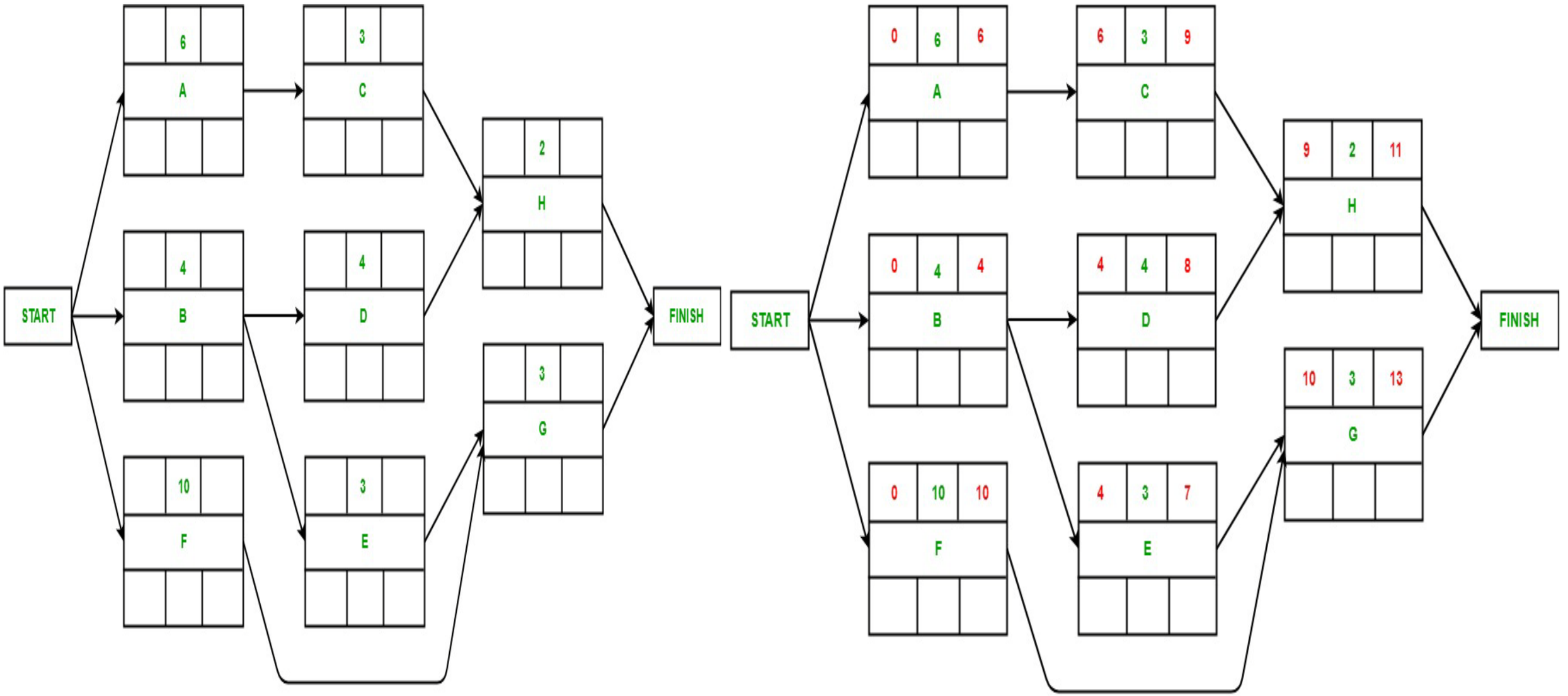
Step 6: Calculate the float

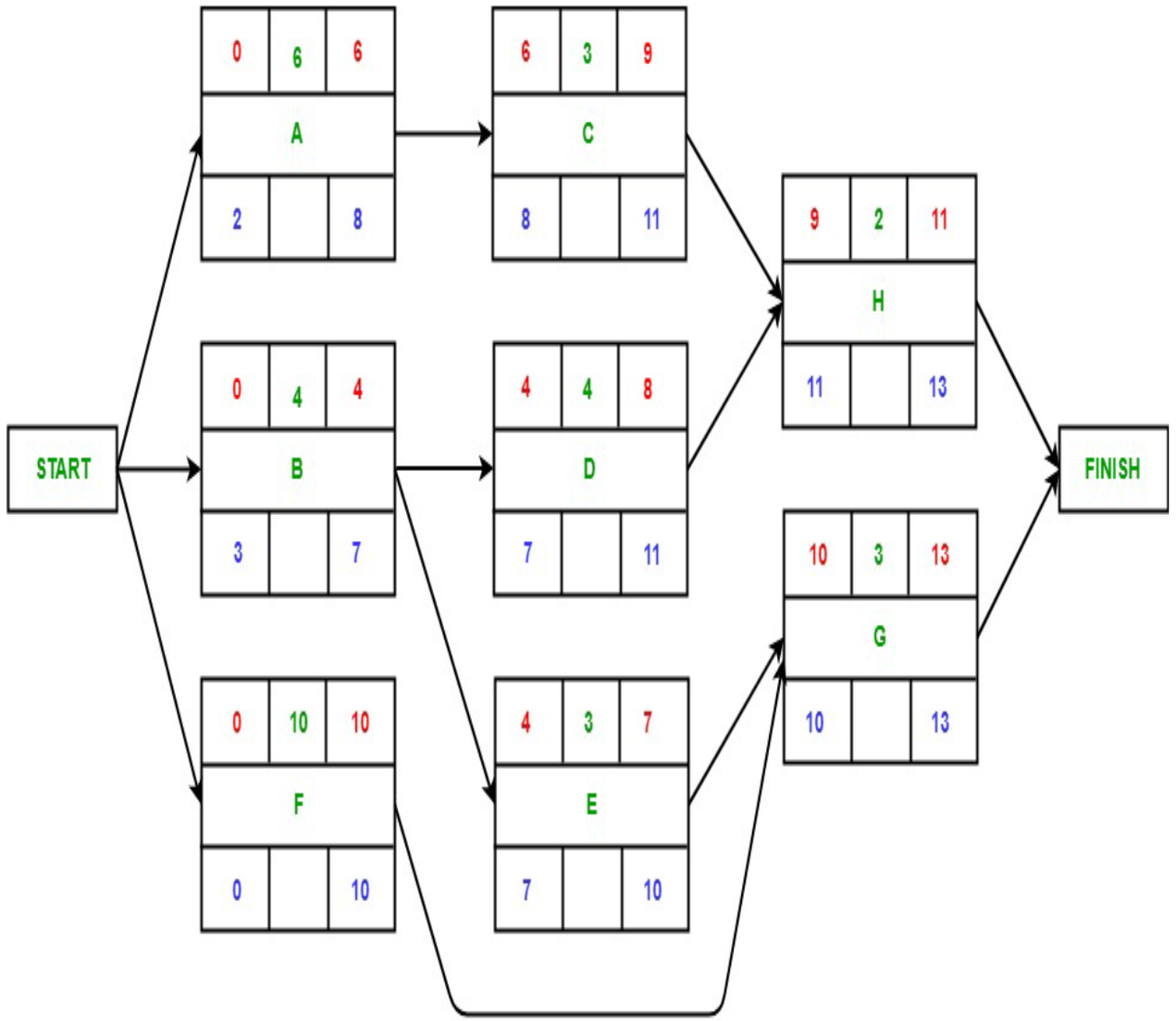
Step 7: Monitor the critical path

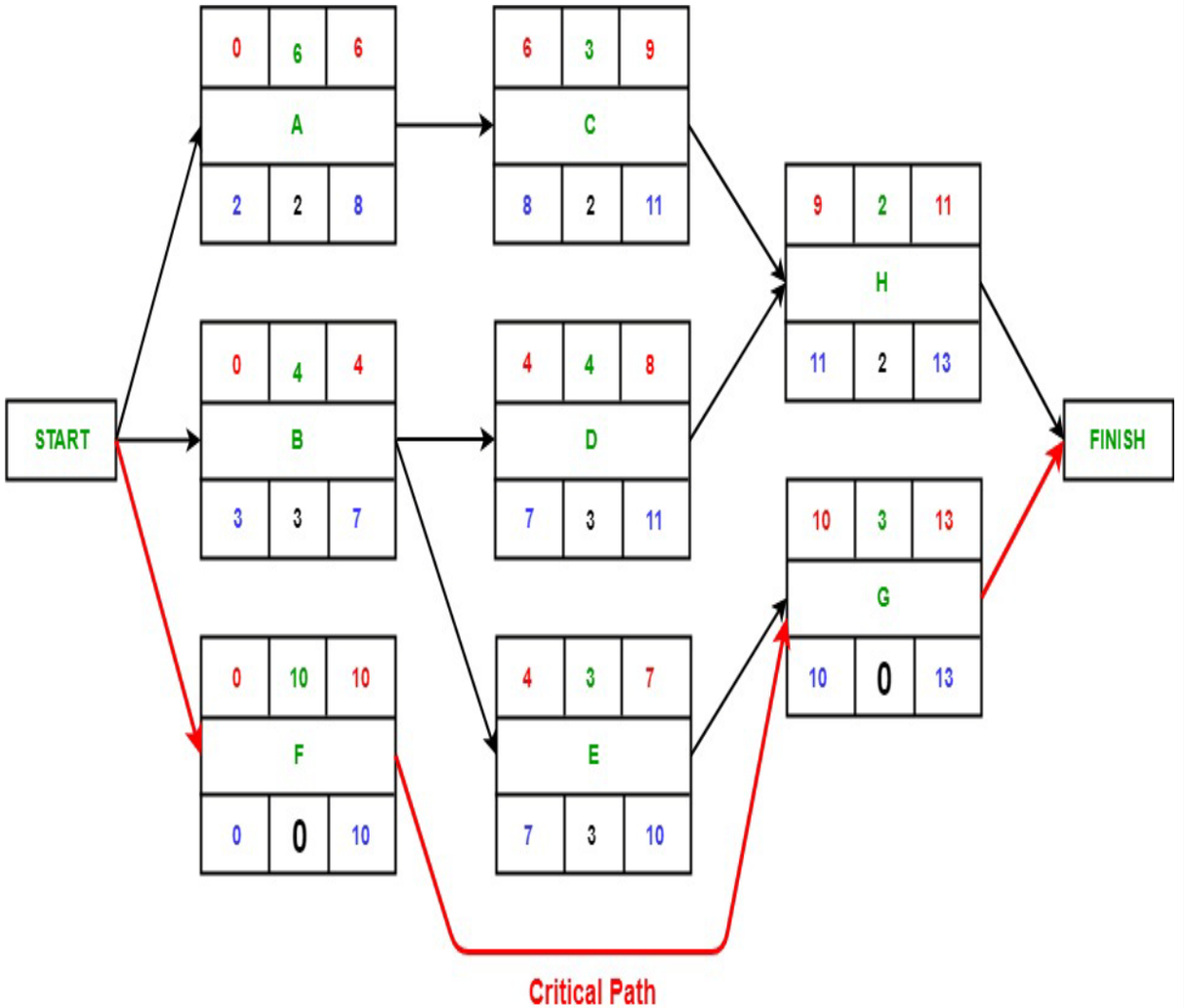
Activity	Duration (in weeks)	Precedents
A	6	–
B	4	–
C	3	A
D	4	B
E	3	B
F	10	–
G	3	E,F
H	2	C,D



- **Activity label** is the name of the activity represented by that node.
- **Earliest Start** is the date or time at which the activity can be started at the earliest.
- **Earliest Finish** is the date or time at which the activity can be completed at the earliest.
- **Latest Start** is the date or time at which the activity can be started at the latest. LS = LF - duration,
- **The latest Finish** is the date or time at which the activity can be finished at the latest.
- **Float** is equal to the difference between the earliest start and latest start or earliest finish and latest finish.
  - Total float = LS - ES or LF - EF


START
A
6
C
3
B
4
D
4
F
10
E
3
H
2
G
3
FINISH
START
A
0
6
6
C
6
3
9
B
0
4
4
D
4
4
8
F
0
10
10
E
4
3
7
H
9
2
11
G
10
3
13
FINISH


START
A
0 6 6
2 8
C
6 3 9
8 11
B
0 4 4
3 7
D
4 4 8
7 11
F
0 10 10
0 10
E
4 3 7
7 10
H
9 2 11
11 13
G
10 3 13
10 13
FINISH


START
A
0 6 6
2 2 8
C
6 3 9
8 2 11
B
0 4 4
3 3 7
D
4 4 8
7 3 11
F
0 10 10
0 0 10
E
4 3 7
7 3 10
H
9 2 11
11 2 13
G
10 3 13
10 0 13
FINISH
Critical Path

Staffing

- Staffing in project management is the process of acquiring, developing, and maintaining a satisfactory and satisfied workforce for a project.
- It involves a series of activities that ensure the organization has the right number of people at the right places, at the right time, and performing the right thing.

Staffing Process

- The process of staffing consists of several interrelated activities, such as planning for human resources requirements, recruitment, selection, training development, remuneration, and so on. These activities together make the staffing process.

Elements or steps of the staffing process

1. Manpower Planning

- Human resource management is a process of determining the number and type of personnel required for filling the vacant job in an organization.
- Manpower requirements involve two kinds of analysis, i.e., **workload analysis** and **workforce analysis.**
- **Workload analysis** involves determining the number and type of employees required to perform various jobs and achieve organizational objectives.
- **Workforce analysis** shows the number and type of human resources available with an organization.

2. Recruitment

- Recruitment refers to a process of searching for prospective employees and encouraging them to apply for jobs in the organization.
- It involves identifying various resources of human force and attracting them to apply for the job.
- The main purpose of a requirement is to create a pool of applicants by a large number of qualified candidates.
- Recruitment can be done by both internal and external sources of recruitment.
- Internal sources may be used to a limited extent, and to get fresh talent and a wider choice, external sources can be used.

3. Selection

- Selection is the process of choosing and appointing the right candidates for various job positions in the organization.
- It is treated as a negative process because it involves the rejection of some candidates.
- The process of selection serves two important purposes, firstly, it ensures that the organization gets the best among the available candidates, and secondly, it boosts ups the self-esteem and prestige of the candidates.

4. Placement and Orientation

- After selection, an appropriate job is assigned to each selected person.
- Placement is the process of matching the candidates with the jobs in the organization.
- Under this process, every selected candidate is assigned a job most suitable for him.
- The purpose of placement is to fit the right person to the right job so that the efficiency of work is high and the employees get personal satisfaction.

5. Training and Development

- People are in search of careers and not jobs.
- Every individual must be given a chance to rise to the top.
- The most favourable way for this to happen is to promote employee learning. For this, organizations either provide training themselves within the organization or through external institutions. This is beneficial for the organization as well.
- If the employees are motivated enough, it will increase their competence and will be able to perform even better for the organization with greater efficiency and productivity.

6. Performance appraisal

- After training the employees and having them on the job for some time, there should be an evaluation done on their performance.
- Every organization has its means of appraisal whether formal or informal.
- Appraisal refers to the evaluation of the employees of the organization based on their past or present performance by some pre-decided standards.

7. Promotion and Career planning

- It has now become important for all organizations to deal with career-related issues and promotional routes for employees.
- The managers should take care of the activities that serve the long-term interests of the employees.
- They should be encouraged from time to time, which will help the employees to grow and find their true potential.
- Promotion refers to the transferring of employees from their current positions to a higher level increasing their responsibilities, authority and pay.

8. Compensation

- Every organization needs to set up plans for the salary and wages of the employees.
- There are several ways to develop payment plans for the employees depending upon the significance of the job.
- All kinds of payments or rewards provided to the employees is referred to as compensation.
- The compensation may be in the form of direct financial payments, such as salary, wages, bonuses, etc., or indirect payments like insurance or vacations provided to the employee.

Software Configuration Management

- Software Configuration Management (SCM) is a discipline within software engineering that focuses on tracking and controlling changes to a software system.
- It encompasses a set of practices and tools that help development teams manage the evolution of their software throughout its lifecycle.
- When we develop software, the product (software) undergoes many changes in their maintenance phase; we need to handle these changes effectively.
- Several individuals (programs) works together to achieve these common goals. This individual produces several work product (SC Items) e.g., Intermediate version of modules or test data used during debugging, parts of the final product.
- The elements that comprise all information produced as a part of the software process are collectively called a software configuration.
- As software development progresses, the number of Software Configuration elements (SCI's) grow rapidly.

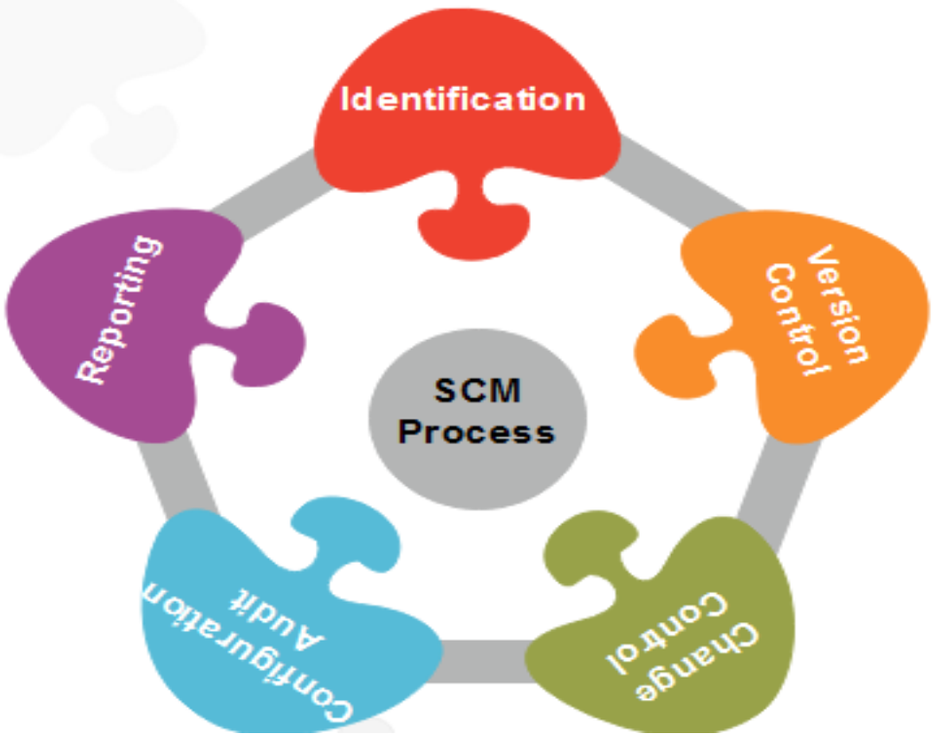
SCM is the discipline which

- Identify change
- Monitor and control change
- Ensure the proper implementation of change made to the item.
- Auditing and reporting on the change made.

SCM Process

- It uses the tools which keep that the necessary change has been implemented adequately to the appropriate component.
- The SCM process defines a number of tasks:

1. Identification of objects in the software configuration
2. Version Control
3. Change Control
4. Configuration Audit
5. Status Reporting



Identification

- Identifying the configuration items from products that compose baselines at given points in time (a baseline is a set of mutually consistent Configuration Items, which has been formally reviewed and agreed upon, and serves as the basis of further development).
- Establishing relationships among items, creating a mechanism to manage multiple levels of control and procedure for the change management system.

Version Control

- Version Control combines procedures and tools to handle different version of configuration objects that are generated during the software process.
- Tracks changes to software code over time, allowing developers to revert to previous versions, compare changes, and collaborate effectively.
- Popular version control systems include Git, SVN, and Mercurial.

Change control

- **Change Control** is a formal process used to manage modifications to a system or project.
- It ensures that any proposed changes are evaluated, approved, and implemented in a controlled manner.
- This helps maintain the integrity and stability of the system while accommodating necessary adjustments.

Configuration Audit

- SCM audits to verify that the software product satisfies the baselines requirements and ensures that what is built and what is delivered.
- SCM audits also ensure that traceability is maintained between all CIs and that all work requests are associated with one or more CI modification.
- SCM audits are the "**watchdogs**" that ensures that the integrity of the project's scope is preserved.

Status Reporting

- Configuration Status reporting (sometimes also called status accounting) providing accurate status and current configuration data to developers, testers, end users, customers and stakeholders through admin guides, user guides, FAQs, Release Notes, Installation Guide, Configuration Guide, etc.