**MODULE- 1**

#### Introduction to software project management -

Software project management is a specialized discipline within the realm of [project management](https://www.simplilearn.com/tutorials/project-management-tutorial/what-is-project-management) that focuses on planning, executing, monitoring, controlling, and closing software development projects. It encompasses the systematic management of resources, budget, and time to ensure the successful completion of a software project. In essence, software project management aims to streamline the complex and multifaceted process of software development by facilitating coordination, communication, and collaboration among team members, stakeholders, and resources involved in the project.

#### Project

A project is a well-defined task, which is a collection of several operations to achieve a goal (for example, software development and delivery). A Project can be characterized as:

* + - Every project may have a unique and distinct goal.
    - The project is not a routine activity or day-to-day operation.
    - The project comes with a start time and end time.
    - Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
    - The project needs adequate resources in terms of time, manpower, finance, material and knowledge bank.

**Software Project**

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve the intended software product.

#### Importance of software project management-

Proper [Software Project Management processes](https://oboloo.com/glossary/software-project-management-process/) are critical for successful completion within scope and budget expectations. With [efficient resource](https://oboloo.com/blog/resource-levelling-in-procurement-ensuring-efficient-allocation-of-resources/) [allocation](https://oboloo.com/blog/resource-levelling-in-procurement-ensuring-efficient-allocation-of-resources/) combined with risk reduction strategies through open communication channels and planned steps taken proactively during development phases; companies can achieve faster turnaround times & cost-efficiency goals. Effective software project management plays a pivotal role in the success of software development endeavors. It is a discipline that ensures the following:

* + - * Efficient Resource Allocation: Proper management of resources, including human resources, finances, time, and technology, helps optimize their utilization, leading to cost-effective and timely project delivery.
      * Clear Communication: Software project management fosters transparent [communication](https://www.simplilearn.com/project-communication-skills-article) channels among team members, stakeholders, and clients. This clarity prevents misunderstandings, promotes alignment, and reduces the likelihood of errors.
      * [Risk Mitigation](https://www.simplilearn.com/what-is-risk-mitigation-article): Identifying and addressing potential [risks](https://www.simplilearn.com/common-project-risks-article) early in the project lifecycle helps minimize their impact and ensure that projects stay on track despite unforeseen challenges.
      * Scope Control: Defining and managing [project scope](https://www.simplilearn.com/project-scope-management-importance-rar89-article) prevents scope creep, ensuring that the project stays focused on its objectives and doesn't deviate unnecessarily, which can lead to time and cost overruns.
      * Quality Assurance: Effective software project management includes quality assurance and testing processes, guaranteeing that the software meets the defined standards and fulfils user requirements.
      * Stakeholder Satisfaction: Involving stakeholders throughout the project makes their expectations and needs more likely to be met, resulting in higher satisfaction levels.
      * Timely Delivery: Through proper scheduling and monitoring, software project management ensures that projects are completed successfully on time, meeting critical deadlines.

#### Key Concepts in Software project Management-

Several key concepts are integral to successful software project management:

* + - * Project Initiation: Defining the project scope, objectives, [stakeholders](https://www.simplilearn.com/stakeholders-impact-on-the-projects-article), and resources required for the project.
      * Project Planning: Creating a [project plan](https://www.simplilearn.com/what-is-a-project-management-plan-article), including tasks, timelines, milestones, resource allocation, and risk assessment.
      * Project Execution: Implementing the project plan, coordinating tasks, managing resources, and ensuring effective communication.
      * Project Monitoring and Control: Tracking project progress, comparing it against the plan, and taking corrective actions when necessary.
      * Risk Management: Identifying potential risks, assessing their impact and probability, and developing strategies to mitigate or respond to them.
      * Quality Management: Ensuring that the software meets defined quality standards and user requirements through testing and validation.
      * Change Management: Managing changes to project scope, requirements, or other aspects while minimizing disruptions.
      * Project Closure: Wrapping up the project, conducting final tests, obtaining approvals, and transitioning the software to its operational phase.

#### Role of Software Project Manager

The software project manager is a central figure in software project management, responsible for overseeing and coordinating all project activities. Their roles and responsibilities include:

* + - * Leadership: Providing vision, direction, and motivation to the project team, guiding them toward achieving project goals.
      * Planning and Organization: Creating comprehensive project plans, allocating resources, and scheduling tasks to ensure efficient project execution.
      * Communication: Promoting transparent and unambiguous communication among stakeholders, clients, and team members to establish a shared understanding.
      * Risk Management: Identifying potential risks, assessing their impact, and devising strategies to mitigate or manage them effectively.
      * Problem Solving: Addressing challenges and issues that arise during the project, making timely decisions to keep the project on track.
      * Monitoring and Control: Regularly monitoring project progress, comparing it to the plan, and taking corrective actions when necessary to ensure the project stays on schedule and within scope.
      * Stakeholder Management: Engaging with stakeholders, understanding their needs, and ensuring their expectations are met throughout the project.
      * Quality Assurance: Implementing quality control processes to ensure the software meets defined quality standards.

#### Software Management Activities

Software project management comprises of a number of activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. Project management activities may include:

#### Project Planning

* + - **Scope Management**

#### Project Estimation Project Planning

Software project planning is task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production. Project planning may include the following:

#### Scope Management

It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product. Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun. During Project Scope management, it is necessary to -

* + - Define the scope
    - Decide its verification and control
    - Divide the project into various smaller parts for ease of management.
    - Verify the scope
    - Control the scope by incorporating changes to the scope

#### Project Estimation

For an effective management accurate estimation of various measures is a must. With correct estimation managers can manage and control the project more efficiently and effectively. Project estimation may involve the following:

#### Software size estimation

Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software. Lines of code depend upon coding practices and Function points vary according to the user or software requirement.

#### Effort estimation

The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers’ experience, organization’s historical data or software size can be converted into efforts by using some standard formulae.

#### Time estimation

Once size and efforts are estimated, the time required to produce the software can be estimated. Efforts required is segregated into sub categories as per the requirement specifications and interdependency of various components of software. Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS). The tasks are scheduled on day-to-day basis or in calendar months. The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

#### Cost estimation

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider

* Size of software
* Software quality
* Hardware
* Additional software or tools, licenses etc.
* Skilled personnel with task-specific skills
* Travel involved
* Communication
* Training and support

**The Triple Constraints of Project Management**



The Triple Constraints of Project Management serve as a model for constraints that come with project management. These three constraints are:

* + - Cost: The [project budget](https://www.simplilearn.com/pmp-examination-preparation-types-of-cost-article), which serves as the financial constraint in a project
    - Scope: The activities necessary to achieve the project's goals
    - Time: The project's schedule based on which the project will be completed

According to the Triple Constraints of Project Management, the project's costs, time, and scope all impact its success. A project manager can maintain control of the triple constraints by balancing these three constraints via tradeoffs.

While the Triple Constraints of Project Management is essential to any [successful project,](https://www.simplilearn.com/how-to-make-a-project-successful-article) it does not define its success. Projects comprise several components, far more than the three that comprise the Triple Constraint. To better reflect the most crucial elements of a project, some project management experts have included these additional limitations to the model:

* Quality: Every project has quality criteria, regardless of whether the end delivery has a tangible or intangible output. To control quality, project managers require a quality management plan.
* Risk: Risk is an unavoidable part of any project. Project managers need to assess and come up with a [risk management](https://www.simplilearn.com/risk-management-strategies-article) plan that estimates and elaborates on how risks would be managed.
* Benefit: Various types of benefits are profited out of projects. A project manager ensures that the best financial benefits are available for project stakeholders.

#### Project Stakeholders

Project Management Institute (2004) defines a project stakeholder as any individual or an organization that is actively involved in a project, or whose interest might be affected as a result of project execution or completion. Some stakeholders are simple to identify. Since the definition includes all people

working in a project, the project manager and the project team, namely, the people responsible for carrying out the work in a project, are stakeholders.

Other stakeholders are those who benefit from the project execution or the project outputs. Among these are the client, the performing organization, and the project sponsor.

The first, in fact, benefits from the project outputs, the second from the know-how and the revenues made in the project, and the third because of the peculiar interest he or she has in the project. The remaining stakeholders might be directly or indirectly affected by the project.

For instance, a company producing a software system might be negatively affected by a project of another organization developing a competing product. Understanding who are the project stakeholders and effectively managing them is an important activity of a project manager.

**External stakeholders** are individuals, groups, or organizations that have an interest in a project but are not directly part of the project team or organization. They often have a significant impact on a project's success and outcomes. Examples of external stakeholders include customers or clients who will use the project's deliverables, suppliers and vendors providing necessary resources, regulators and government agencies responsible for industry standards and compliance, and even competitors who may be affected by or interested in the project's results. Engaging effectively with external stakeholders is essential to align the project with their expectations and needs.

**Internal stakeholders** are individuals or groups within an organization who have a direct interest in the success of a particular project. They often play critical roles in project planning, execution, and decision-making. Examples of internal stakeholders include project sponsors, project managers, project team members, and functional managers. Internal stakeholders are typically more directly involved in the project's day-to-day activities and have a vested interest in achieving the project's objectives. Effective communication and collaboration with internal stakeholders are essential for project success.

#### Ways of categorizing software projects

Different characteristics of a project could affect the way in which it should be planned and managed. Some of these are: –

* + Compulsory vs voluntary users –
  + Information systems vs embedded systems –
  + Software products vs software services –
  + Objective driven development vs product driven development.

#### Compulsory vs. Voluntary Users:

* + - **Compulsory Users**: These are users who have no choice but to use the software. For example, government tax software is used by citizens because it's mandatory. Projects with compulsory users may have strict regulatory requirements and a focus on compliance and security.
    - **Voluntary Users**: These are users who choose to use the software, often for personal or business reasons. Social media platforms, entertainment apps, and productivity software fall into this category. Projects with voluntary users may focus more on user experience and innovation to attract and retain users.

#### Information Systems vs. Embedded Systems:

* + - **Information Systems**: These software projects primarily deal with managing and processing data and information. Examples include customer relationship management (CRM) systems, enterprise resource planning (ERP) software, and database management systems. These projects often involve extensive data handling and integration.
    - **Embedded Systems**: These projects involve software that is tightly integrated into hardware, such as in automotive control systems, medical devices, or smart appliances. Embedded systems require real-time performance, reliability, and hardware-software co-design.

#### Software Products vs. Software Services:

* + - **Software Products**: These projects focus on developing standalone software applications or packages that are sold or distributed to end-users. Examples include operating systems, productivity software, and video games. Product development may emphasize market competitiveness, usability, and feature richness.
    - **Software Services**: These projects involve creating software services or platforms that are accessed over the internet. Examples include cloud computing services, web-based applications, and online collaboration

tools. Service projects may prioritize scalability, availability, and a subscription-based revenue model.

#### Objective-Driven Development vs. Product-Driven Development:

* + - **Objective-Driven Development**: In this approach, software projects are initiated to achieve specific goals or objectives. For example, a company might develop software to streamline its supply chain operations or improve customer support. These projects are often driven by immediate business needs.
    - **Product-Driven Development**: In this approach, the focus is on creating software products for a broader market. The primary goal is to develop a software product that can be marketed and sold to a wide range of customers. This often involves market research, feature prioritization, and a longer-term strategic vision.

#### Problems with Software Project Management

[Project managers](https://synoptek.com/insights/it-blogs/project-management-best-practices-drive-successful-software-development/) are always under immense pressure to deliver projects on time, within budget, and as per expectations. Yet, in most cases, they are met with several issues all along the lifecycle. Although the potential issues associated with any project may grow up to an exhaustive number of items depending upon the type, scope, and other attributes of the project, here are 6 problems that are most widespread:

* + - Not Enough Planning Ahead

In the race to bring products into the market quickly, many teams compromise on the planning phase, often not realizing that proper planning lies at the crux of every successful project. If you want your project to meet its goals, planning ahead is the most important factor in getting it off the ground. Irrespective of time-to-market pressures, make sure to take out time in planning the scope, budget, resources, and tools.

* + - Setting the Bar Too High

When teams begin working on a project, they tend to set the bar too high – often aspiring to achieve milestones that are impractical. Although it is good to want the product to meet several specifications, setting out with expectations that are

too optimistic often leads to setting them up for failure. When beginning to work on any project, make sure to set SMART goals; goals you can easily measure and achieve with the tools and resources you have in the specified frame of time.

* + - Keeping up with the Latest Market Trends

Given how complex today’s projects have become, they often span several months or even years. It is very likely for technologies and trends to change during the course of the project. As third-party technologies soar, teams may be left wondering how to adapt – which may throw a wrench in your project. Keeping up with the latest trends requires teams to [embrace agile](https://synoptek.com/insights/it-blogs/agile-scrum-waterfall-software-development-difference/) and micro services-based approaches, so new technologies and trends can easily be incorporated into the ongoing project.

* + - Facing the Skills Gap

With software forming the foundation of almost every business today, the skills gap has become increasingly wide. As more and more projects need to be kicked off, many organizations are having trouble attracting and retaining the right talent. Overcoming the skills gap requires project managers to engage with qualified IT providers on a pro-project basis and meet their project goals with ease.

* + - Missing Requirements

Another challenge that rise with today’s projects is improper requirements gathering that could lead to issues later down the road. Building products with unspecific or oscillating requirements can cause scope creep, overshooting budgets, as well as missed deadlines. To set the project for optimal success, it is important for project managers to identify stakeholders, gather and document key requirements, and implement a well-planned requirements gathering process.

* + - Engaging C-suite and Stakeholders

No matter the pressure project managers are in, you must educate high-level executives and stakeholders in order to secure buy-in. Engaging the C-suite early in the [project lifecycle](https://synoptek.com/insights/it-blogs/what-is-software-development-life-cycle-and-where-do-i-start/) ensures you have the approvals you need across budget, tools, and resources. Their support also helps in overcoming any issues you might face as you bring your project to closure.

#### NOTE - "C-suite" is the widely used term that describes an organization's senior executives. The titles of senior executives often start with the letter “C,” such as a chief executive officer or chief financial officer.

**Phases of project management lifecycle**

The project management lifecycle is a step-by-step framework of best practices used to monitor a project from its beginning to its end. It provides project managers with a structured way to create, execute, and finish a project. This project management process generally includes four phases: initiating, planning, executing, and closing. Some may also include a fifth “monitoring and controlling” phase between the executing and closing stages. Each step plays a crucial role in making sure the project has the best chance of achieving its goals. The project management lifecycle provides projects with structure and tools to ensure they have the best chance of being successful. As a project manager, it is a process you will want to know well.

#### The Project Management Lifecycle: 4 Steps

1. **Initiating**

In the initiation phase, you will define the project. You will sort out the project goals, scope, and resources of the project and what roles are needed on the team. Clarifying what stakeholders expect out of the project, and what exactly the project is aiming to achieve (and why) will give the project and team clear direction. This is a crucial phase to the project’s success. Without clarity around what needs to be achieved and why the project runs the risk of not accomplishing the end results and meeting the expectations of stakeholders.

Some steps in the initiation phase include:

* Communicating with stakeholders to understand the purpose and desired outcomes of the project
* Identifying the scope of the project
* Determining SMART goals (specific, measurable, achievable, relevant, and time-bound)
* Clarifying resources like budget and time constraints
* Confirming team size and roles required
* Determining how often and which stakeholders will be involved throughout the project
* Compiling a project proposal and project charter

Tools and documents used in the initiation phase can include:

* **Project proposal:** The project proposal defines a project and outlines key dates, requirements, and goals.
* **Project charter:** This is a definitive document that describes the project and main details necessary to reach its goals. This can include potential risks, benefits, constraints, and key stakeholders.
* **RACI chart:** A RACI chart, lays down the roles and responsibilities of members of a project team.

#### Planning

In the planning phase, you will determine the steps to actually achieve the project goals—the “how” of completing a project.

You will establish budgets, timelines, milestones, source materials, and necessary documents. This step also involves calculating and predicting risk, establishing change processes in place, and outlining communication protocols. If the initiation phase is assembling your troops, the planning phase is deciding what to do with them.

The planning phase can include the following steps:

* Deciding on milestones that lead up to goal completion
* Developing a schedule for tasks and milestones, including time estimates and potential time buffers
* Establishing change processes
* Determining how and how often to communicate with team members and stakeholders
* Creating and signing documents such as non-disclosure agreements (NDAs) or requests for proposal (RFPs)
* Assessing and managing risk by creating a risk register
* Holding a kick-off meeting to start a project Tools you might use in a planning phase include:
* **Gantt chart:** A horizontal bar chart in which members can see what tasks must be completed in which order and how long each task is expected to take
* **Risk register:** A chart that lists risks associated with the project, along with their probability, potential impact, risk level, and mitigation plans

#### Execute and complete tasks

Executing a project means putting your plan into action and keeping the team on track. Generally, this means tracking and measuring progress, managing quality, mitigating risk, managing the budget, and using data to inform your decisions.

Specific steps might include:

* + Using tools like GANTT or burn down charts to track progress on tasks
  + Responding to risks when they manifest
  + Recording costs
  + Keeping team members motivated and on task
  + Keeping stakeholders informed of progress
  + Incorporating changes via change requests

Some tools you might use include:

* **Change requests:** These are documents used to propose changes to a project’s scope or goals
* **Burn down chart:** This chart breaks down tasks on a granular level and visualize the amount of time remaining

#### Close projects

In the closing phase of the project management lifecycle, you will conclude project activities, turn the finished product or service over to its new owners and assess the things that went well and did not go so well. It will also be a time to celebrate your hard work.

Steps in the closing phase can include:

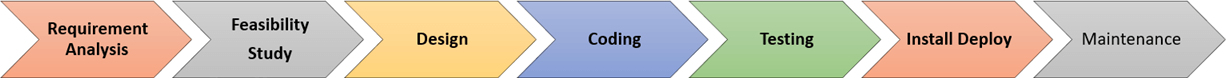
1. Conducting retrospectives and take notes of changes you can implement in the future
2. Communicating to stakeholders at the end of the project and providing an impact report
3. Communicating with the new owners of a project
4. Creating a project closeout report
5. Celebrating the end of the project and your successes
6. Tools used in the closing phase include:
7. **Impact report:** This report compiles a series of metrics that showcase how your project made a difference and is presented to your stakeholders.
8. **Project closeout report:** A project closeout report provides a summary of your project’s accomplishments, and provides key learning for future project managers to reference.

#### Software Development Life Cycle (SDLC) Phases What is SDLC?

**SDLC** is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the pre-defined time frame and cost. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC life Cycle has its own process and deliverables that feed into the next phase. SDLC stands for **Software Development Life Cycle** and is also referred to as the Application Development life-cycle.

#### SDLC Phases

The entire SDLC process divided into the following SDLC steps:



SDLC Phases

* Phase 1: Requirement collection and analysis
* Phase 2: Feasibility study
* Phase 3: Design
* Phase 4: Coding
* Phase 5: Testing
* Phase 6: Installation/Deployment
* Phase 7: Maintenance

#### Phase 1: Requirement collection and analysis

Requirement collection and analysis is the foundational phase in software development, where project teams engage with stakeholders to gather and document the needs and expectations for the software. It involves eliciting, documenting, and analyzing requirements, prioritizing them, and validating their accuracy. Stakeholder validation and requirements sign-off are crucial steps to ensure alignment with project objectives. This phase establishes a baseline for the development process and forms the basis for making informed decisions and managing changes effectively throughout the project's lifecycle.

#### Phase 2: Feasibility study

This stage involves a thorough assessment of the project's technical, operational, and financial feasibility. Technical feasibility examines whether the project can be executed with available technology and expertise. Operational feasibility assesses the software's impact on current operations. Financial feasibility analyzes the project's budget and potential return on investment. The outcome is a feasibility report that offers recommendations on whether to proceed with the project. This phase is essential in helping organizations make well-informed decisions, ensuring that the project aligns with their capabilities and financial constraints, and has a reasonable chance of success.

#### Phase 3: Design

This stage involves creating detailed technical specifications for the software based on the gathered requirements. It encompasses system architecture, database design, user interface design, and technical documentation. The design phase provides a clear blueprint for the development team to follow, ensuring that the software will be well-structured and efficient. Additionally, it lays the foundation for quality assurance planning and serves as a crucial reference throughout the development process, guiding the implementation of the software in a systematic and organized manner.

#### Phase 4: Coding

In this phase developers write the actual code for the software based on the design specifications. This phase involves the practical implementation of the software's functionality, making use of chosen programming languages and technologies. Developers work on creating modular, well-structured code, often undergo peer code reviews, and follow coding standards. Testing is initiated to catch and rectify issues early, and documentation is essential for understanding and maintaining the code.

#### Phase 5: Testing

During this stage, the software is systematically evaluated to identify and rectify defects, ensuring that it meets the specified requirements. Various types of testing are conducted, including unit testing, integration testing, system testing, and user acceptance testing, each focusing on different aspects of the software.

* ***Unit testing*** is a fundamental testing technique in software development. It focuses on verifying the correctness and functionality of individual components or units of code in isolation.
* ***Integration testing*** is a phase of software testing that focuses on verifying the interactions and interfaces between various components or units of a software system.
* ***System testing*** is a critical phase where the entire software system is tested as a whole to ensure that it meets the specified requirements and functions as expected.
* ***User Acceptance Testing*** is the final phase of the software testing process, where end-users or client representatives validate that the software meets their requirements and is ready for deployment.

#### Phase 6: Installation/Deployment

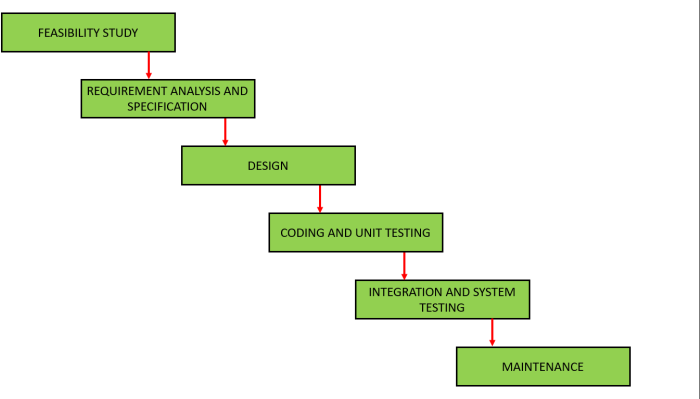
The installation/deployment phase is where the software becomes operational, and end-users start using it to perform their tasks. It is crucial to ensure a smooth transition from development to production while minimizing disruptions and ensuring that the software functions as intended in its live environment.

#### Phase 7: Maintenance

The maintenance phase is essential to keep the software reliable, secure, and aligned with evolving user requirements. It ensures the software remains a valuable and sustainable asset for the organization over time.

#### Waterfall Model

The classical waterfall model is the basic [software development life](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/) [cycle](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/) model. It is very simple but idealistic. Earlier this model was very popular but nowadays it is not used. But it is very important because all the other software development life cycle models are based on the classical waterfall model.



#### Feasibility Study

The main goal of this phase is to determine whether it would be financially and technically feasible to develop the software. The feasibility study involves understanding the problem and then determining the various possible strategies to solve the problem. These different identified solutions are analyzed based on their benefits and drawbacks, The best solution is chosen and all the other phases are carried out as per this solution strategy.

#### Requirements Analysis and Specification

The aim of the requirement analysis and specification phase is to understand the exact requirements of the customer and document them properly. This phase consists of two different activities.

* + **Requirement gathering and analysis:** Firstly all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed. The goal of the analysis part is to remove incompleteness (an incomplete requirement is one in which some parts of the actual requirements have been omitted) and inconsistencies (an inconsistent requirement is one in which some part of the requirement contradicts some other part).
  + **Requirement specification:** These analyzed requirements are documented in a software requirement specification (SRS) document. SRS document serves as a contract between the development team and customers. Any future dispute between the customers and the developers can be settled by examining the SRS document.

#### Design

The goal of this phase is to convert the requirements acquired in the SRS into a format that can be coded in a programming language. It includes high-level and detailed design as well as the overall software architecture. A [Software Design](https://www.geeksforgeeks.org/design-documentation-in-software-engineering/) [Document](https://www.geeksforgeeks.org/design-documentation-in-software-engineering/) is used to document all of this effort (SDD)

#### Coding and Unit Testing

In the coding phase software design is translated into source code using any suitable programming language. Thus each designed module is coded. The aim of the unit testing phase is to check whether each module is working properly or not.

#### Integration and System testing

Integration of different modules is undertaken soon after they have been coded and unit tested. Integration of various modules is carried out incrementally over a number of steps. During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this. System testing consists of three different kinds of testing activities as described below.

* + **Alpha testing:** Alpha testing is the system testing performed by the development team.
  + **Beta testing:** Beta testing is the system testing performed by a friendly set of customers.
  + **Acceptance testing:** After the software has been delivered, the customer performed acceptance testing to determine whether to accept the delivered software or reject it.

#### Maintenance

Maintenance is the most important phase of a software life cycle. The effort spent on maintenance is 60% of the total effort spent to develop a full software. There are basically three types of maintenance.

* + **Corrective Maintenance:** This type of maintenance is carried out to correct errors that were not discovered during the product development phase.
  + **Perfective Maintenance:** This type of maintenance is carried out to enhance the functionalities of the system based on the customer’s request.
  + **Adaptive Maintenance:** Adaptive maintenance is usually required for porting the software to work in a new environment such as working on a new computer platform or with a new operating system.

#### An overview of Project Planning

Once a project is found to be possible, computer code project managers undertake project designing. Project designing is undertaken and completed even before any development activity starts. Project designing consists of subsequent essential activities:

Estimating the subsequent attributes of the project:

#### Project Size:

What’s going to be downside quality in terms of the trouble and time needed to develop the product?

#### Cost:

What proportion is it reaching to value to develop the project?

#### Duration:

However long is it reaching to want complete development?

#### Effort:

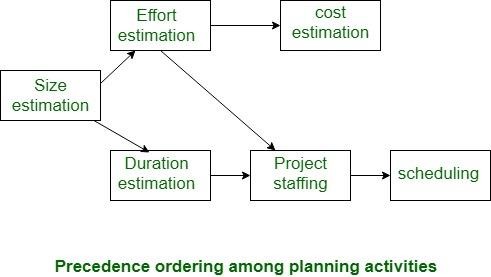
What proportion effort would be required?

The effectiveness of the following designing activities relies on the accuracy of those estimations.

* + planning force and alternative resources
  + workers organization and staffing plans
  + Risk identification, analysis, and abatement designing
  + Miscellaneous arranges like quality assurance plan, configuration, management arrange, etc.

#### Precedence ordering among project planning activities:

The different project connected estimates done by a project manager have already been mentioned. The below diagram shows the order during which vital project coming up with activities is also undertaken. It may be simply discovered that size estimation is that the 1st activity. It’s conjointly the foremost basic parameter supported that all alternative coming up with activities square measure dispensed, alternative estimations like the estimation of effort, cost, resource, and project length also are vital elements of the project coming up with.



#### Sliding Window Planning:

Project designing needs utmost care and a spotlight since commitment to unrealistic time and resource estimates end in schedule slippage. Schedule delays will cause client discontent and adversely have an effect on team morale. It will even cause project failure.

However, project designing could be a terribly difficult activity. particularly for giant comes, it’s pretty much troublesome to create correct plans. A region of this issue is thanks to the actual fact that the correct parameters, the scope of the

project, project workers, etc. might amendment throughout the span of the project. So as to beat this drawback, generally project managers undertake project designing little by little. Designing a project over a variety of stages protects managers from creating huge commitments too early. This method of staggered designing is thought of as window designing. Within the window technique, beginning with associate initial set up, the project is planned additional accurately in sequential development stages.

At the beginning of a project, project managers have incomplete information concerning the main points of the project. Their info base step by step improves because the project progresses through completely different phases.

When the completion of each section, the project managers will set up every ulterior section additional accurately and with increasing levels of confidence.

#### Project Evaluation

Project evaluation is the process of measuring the success of a project, program or [portfolio](https://www.projectmanager.com/guides/project-portfolio-management). This is done by gathering data about the project and using an evaluation method that allows evaluators to find performance improvement opportunities. Project evaluation is also critical to keep stakeholders updated on the project status and any changes that might be required to the budget or schedule.

Project evaluation can be broken down into three main types or methods: pre- project evaluation, ongoing evaluation and post-project evaluation.

#### Pre-Project Evaluation

In a sense, you’re pre-evaluating your project when you [write your project](https://www.projectmanager.com/blog/project-charter) [charter](https://www.projectmanager.com/blog/project-charter) to pitch to the stakeholders. You cannot effectively plan, staff and control a new project if you’ve first not evaluated it. Pre-project evaluation is the only sure way you can determine the effectiveness of the project before executing it.

#### Ongoing Project Evaluation

To make sure your project is proceeding as planned and hitting all of the scheduling and budget milestones you’ve set, it’s crucial that you constantly [monitor and report on your work in real-time.](https://www.projectmanager.com/project-tracker) Only by using project metrics can you measure the success of your project and whether or not you’re meeting the project’s goals and objectives. It’s strongly recommended that you use project management dashboards and tracking tools for ongoing evaluation.

#### Post-Project Evaluation

Think of this as a postmortem. Post-project evaluation is when you go through the project’s paperwork, interview the [project team](https://www.projectmanager.com/blog/assemble-a-project-team) and principles and analyze all relevant data so you can understand what worked and what went wrong. Only by developing this clear picture can you resolve issues in upcoming projects.

**Selection of appropriate project approach**

Right approach to a particular project: variously called technical planning, project analysis, methods engineering and methods tailoring

* In-house: means that the developers and the users of the software are in the same organization. Often the methods to be used dictated by organizational standards
* Suppliers: means that the developers and the users of the software are in the different organization. It is needed for tailoring as different customers have different needs.

**Developing a new IT application in-house:**

* Time is needed to develop the software.
* Would often require the recruitment of new technical staff to do the job
* Usually, the new staff won’t be needed after the project is completed
* Sometimes due to the novelty of the project there may be lack of executives to lead the effort

**Contracting the project out to an external IT development company (Outsourcing):**

* Time is needed to develop the software
* The conducting company will have technical and project expertise not readily available to the client
* The client would still do management effort to establish and manage the contracts

**Project size estimation techniques**

Project size estimation is a crucial aspect of software engineering, as it helps in planning and allocating resources for the project. Here are some of the popular project size estimation techniques used in software engineering:

**Expert Judgment:** In this technique, a group of experts in the relevant field estimates the project size based on their experience and expertise. This technique is often used when there is limited information available about the project.

**Analogous Estimation**: This technique involves estimating the project size based on the similarities between the current project and previously completed projects. This technique is useful when historical data is available for similar projects.

**Bottom-up Estimation:**  In this technique, the project is divided into smaller modules or tasks, and task is estimated separately each. The estimates are then aggregated to arrive at the overall project estimate.

**Three-point Estimation:** This technique involves estimating the project size using three values: optimistic, pessimistic, and most likely. These values are then used to calculate the expected project size using a formula such as the PERT formula.

**Function Points:** This technique involves estimating the project size based on the functionality provided by the software. Function points consider factors such as inputs, outputs, inquiries, and files to arrive at the project size estimate.

**Use Case Points:** This technique involves estimating the project size based on the number of use cases that the software must support. Use case points consider factors such as the complexity of each use case, the number of actors involved, and the number of use cases.

    Each of these techniques has its strengths and weaknesses, and the choice of technique depends on various factors such as the project’s complexity, available data, and the expertise of the team.

Estimation of the size of the software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:

* Lines of Code
* Number of entities in ER diagram
* Total number of processes in detailed data flow diagram
* Function points

1. **Lines of Code (LOC):**

As the name suggests, LOC counts the total number of lines of source code in a project. The units of LOC are:

* KLOC- Thousand lines of code
* NLOC- Non-comment lines of code
* KDSI- Thousands of delivered source instruction

The size is estimated by comparing it with the existing systems of the same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.

It’s tough to estimate LOC by analyzing the problem definition. Only after the whole code has been developed can accurate LOC be estimated. This statistic is of little utility to project managers because project planning must be completed before development activity can begin.

Two separate source files having a similar number of lines may not require the same effort. A file with complicated logic would take longer to create than one with simple logic. Proper estimation may not be attainable based on LOC.

The length of time it takes to solve an issue is measured in LOC. This statistic will differ greatly from one programmer to the next. A seasoned programmer can write the same logic in fewer lines than a newbie coder.

.

1. **Number of entities in ER diagram:**

[ER model](https://www.geeksforgeeks.org/database-management-system-er-model/) provides a static view of the project. It describes the entities and their relationships. The number of entities in ER model can be used to measure the estimation of the size of the project. The number of entities depends on the size of the project. This is because more entities needed more classes/structures thus leading to more coding.

1. **Total number of processes in detailed data flow diagram:**

Data Flow Diagram(DFD) represents the functional view of software. The model depicts the main processes/functions involved in software and the flow of data between them. Utilization of the number of functions in DFD to predict software size. Already existing processes of similar type are studied and used to estimate the size of the process. Sum of the estimated size of each process gives the final estimated size.

1. **Function Point Analysis:**

In this method, the number and type of functions supported by the software are utilized to find FPC(function point count). The steps in function point analysis are:

* Count the number of functions of each proposed type.
* Compute the Unadjusted Function Points(UFP).
* Find the Total Degree of Influence(TDI).
* Compute Value Adjustment Factor(VAF).
* Find the Function Point Count(FPC).

The explanation of the above points is given below:

* **Count the number of functions of each proposed type:** Find the number of functions belonging to the following types:
  + External Inputs: Functions related to data entering the system.
  + External outputs: Functions related to data exiting the system.
  + External Inquiries: They lead to data retrieval from the system but don’t change the system.
  + Internal Files: Logical files maintained within the system. Log files are not included here.
  + External interface Files: These are logical files for other applications which are used by our system.
* **Compute the Unadjusted Function Points(UFP):** Categorise each of the five function types like simple, average, or complex based on their complexity. Multiply the count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are as follows:

| **Function type** | **Simple** | **Average** | **Complex** |
| --- | --- | --- | --- |
| External Inputs | 3 | 4 | 6 |
| External Output | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |
| Internal Logical Files | 7 | 10 | 15 |
| External Interface Files | 5 | 7 | 10 |

* **Find Total Degree of Influence:** Use the ’14 general characteristics’ of a system to find the degree of influence of each of them. The sum of all 14 degrees of influence will give the TDI. The range of TDI is 0 to 70. The 14 general characteristics are: Data Communications, Distributed Data Processing, Performance, Heavily Used Configuration, Transaction Rate, On-Line Data Entry, End-user Efficiency, Online Update, Complex Processing Reusability, Installation Ease, Operational Ease, Multiple Sites and facilitate Change.

Each of the above characteristics is evaluated on a scale of 0-5.

* **Compute Value Adjustment Factor(VAF):** Use the following formula to calculate VAF

VAF = (TDI \* 0.01) + 0.65

* **Find the Function Point Count:** Use the following formula to calculate FPC

FPC = UFP \* VAF

**Effort Estimation**

Estimating efforts in software engineering is a critical aspect of product development. Accurate effort estimation helps teams plan effectively, allocate resources efficiently, and meet project deadlines. However, estimating efforts in software engineering can be challenging due to the inherent complexity and uncertainty involved in the development process. Today we will explore some key strategies and techniques to help you estimate efforts more effectively in software engineering during product development.

1**. Break Down the Project:** To estimate efforts accurately, it is important to break down the project into smaller, more manageable tasks. By breaking down the project, you can identify the specific activities and deliverables involved. This breakdown allows for a more granular estimation of efforts, enabling better planning and resource allocation.

2.**Consider Complexity and Dependencies**: Software development projects often involve various complexities and dependencies. Analyze the complexity of the tasks and the interdependencies between them. Tasks that require integration with external systems, complex algorithms, or specialized skills will generally require more effort. Consider these factors when estimating efforts for each task.

3.**Leverage Historical Data:** Historical data from past projects is a valuable resource for estimating efforts. Review similar projects completed in the past and analyze the effort, time, and resources required. Look for patterns and commonalities between the current project and previous ones to make informed estimates. Historical data provides a benchmark that can guide your estimation process.

4.**Involve the Development Team:** Collaboration with your development team is crucial for accurate effort estimation. Engage developers and domain experts in the estimation process. Their technical expertise and experience can help you identify potential challenges, risks, and considerations that may impact effort estimation. Including the team in the estimation process also improves buy-in and commitment to the estimates.

**5.Use Estimation Techniques**: Various estimation techniques are available in software engineering. One popular approach is the use of estimation techniques such as Function Point Analysis (FPA) or Use Case Points (UCP). These techniques use mathematical formulas and data points to estimate efforts based on the functionality or use cases of the software. These methods provide a structured and systematic approach to estimation.

**6. Consider Risks and Uncertainties**: Software development projects are often subject to uncertainties and risks. It is crucial to identify and assess potential risks that could impact the effort estimation process. Consider factors such as new technologies, dependencies on third-party components, and potential changes in requirements. Allocate contingency time for handling unforeseen risks and uncertainties.

**7.Regularly Review and Refine Estimates:** Effort estimation is not a one-time task. It requires continuous monitoring and refinement as the project progresses. Regularly review your estimates in light of the actual progress and adjust them if necessary. Learn from past estimation experiences and incorporate feedback to improve future estimates.

8.**Document Assumptions and Rationale:** Documenting the assumptions made during the estimation process and the rationale behind the estimates is essential. This documentation provides transparency and helps stakeholders understand the basis of the effort estimates. It also serves as a reference point for future projects and enables you to learn from previous experiences.

Effort estimation in software engineering is a complex task, but by following these strategies and techniques, you can improve the accuracy of your estimates. Remember to involve the development team, leverage historical data, break down the project, and consider risks and uncertainties. Effort estimation is an ongoing process that requires continuous improvement and refinement. With practice and experience, you can enhance your estimation skills and contribute to more successful software product development.

**COCOMO MODEL-**

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e **number of Lines of Code**. It 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. It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects, which makes it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

* **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
* **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, and months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of the constant to be used in subsequent calculations.

These characteristics pertaining to different system types are mentioned below. Boehm’s definition of organic, semidetached, and embedded systems:

**1. Organic –** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

**2. Semi-detached –** A software project is said to be a Semi-detached type if the vital characteristics such as team size, experience, and knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered Semi-Detached types.

**3. Embedded –** A software project requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

**1.Basic Model –**

E= a(KLOC)^b

time= c(Effort)^d

Person required = Effort/ time

The above formula is used for the cost estimation of for the basic COCOMO model, and also is used in the subsequent models. The constant values a,b,c, and d for the Basic Model for the different categories of the system:

The effort is measured in Person-Months and as evident from the formula is dependent on Kilo-Lines of code. The development time is measured in months. These formulas are used as such in the Basic Model calculations, as not much consideration of different factors such as reliability, and expertise is taken into account, henceforth the estimate is rough.

**2. Intermediate Model –** The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software systems. However, in reality, no system’s effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, and Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation. Classification of Cost Drivers and their Attributes:

**(i) Product attributes –**

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

**3. Detailed Model –** Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver’s impact on each step of the software engineering process. The detailed model uses different effort multipliers for each cost driver attribute. In detailed cocomo, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort. The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

**Advantages of the COCOMO model:**

1. Provides a systematic way to estimate the cost and effort of a software project.
2. Can be used to estimate the cost and effort of a software project at different stages of the development process.
3. Helps in identifying the factors that have the greatest impact on the cost and effort of a software project.
4. Can be used to evaluate the feasibility of a software project by estimating the cost and effort required to complete it.

**Disadvantages of the COCOMO model:**

1. Assumes that the size of the software is the main factor that determines the cost and effort of a software project, which may not always be the case.
2. Does not take into account the specific characteristics of the development team, which can have a significant impact on the cost and effort of a software project.
3. Does not provide a precise estimate of the cost and effort of a software project, as it is based on assumptions and averages.

**Halstead metrics-**

Halstead metrics, named after Maurice H. Halstead, are a set of software metrics used to measure various aspects of a software program. These metrics are primarily used to gauge the complexity and size of a program and were introduced by Halstead in the 1970s.

There are four primary metrics in Halstead's software science:

1. **Program Length (N):** This metric represents the total number of operators and operands in a program. It includes both unique operators and unique operands.
2. **Program Vocabulary (n):** This metric represents the total number of unique operators and operands in a program. It counts each unique operator and operand only once.
3. **Program Volume (V):** The program volume is a measure of the size or length of a program and is calculated using the formula:

*V*=*N*×log2​(*n*)

It provides an indication of the program's complexity. Higher program volume values suggest higher complexity.

1. **Program Difficulty (D):** The program difficulty is calculated using the formula:

*D*=*n*1​(2*N*​)(*Nn*​)

It is a measure of how hard it is to understand a program. Lower difficulty values indicate easier-to-understand programs.

In addition to these primary metrics, there are derived metrics such as Effort and Time Estimation, which are calculated based on the primary metrics. It's worth noting that while Halstead metrics were influential and widely used in the past, they have some limitations and may not capture all aspects of software complexity. Modern software engineering practices often use a combination of various metrics and methodologies to assess software quality and maintainability.

**MODULE-2**

#### Activity Planning-

Activity planning in software project management involves the identification, sequencing, and scheduling of tasks essential for successful project completion. This process typically begins with a thorough breakdown of project requirements into specific activities, creating a Work Breakdown Structure (WBS). Activities are then sequenced to establish dependencies and relationships, ensuring a logical flow of work. Once the sequence is determined, project managers assign resources, estimate time durations, and create a project schedule. This schedule serves as a roadmap, guiding the team through the project phases and milestones. Effective activity planning is crucial for resource optimization, risk management, and timely delivery of a high-quality software product, aligning team efforts with project goals and client expectations. Regular monitoring and adjustment of the project schedule during its execution are essential to accommodate unforeseen challenges and changes, ensuring the project stays on track and meets its objectives. Activity planning is an iterative process, and adjustments may be necessary as the project progresses. Effective planning is crucial for successful project execution and delivery.

The objective of activity planning in software project management is to:

1. Breakdown Project Work: Divide the project into manageable tasks and activities.
2. Sequence Tasks: Determine the order and dependencies of tasks for efficient execution.
3. Estimate Time and Resources: Assess the time and resources required for each task.
4. Create a Schedule: Develop a timeline that outlines when each task will be performed.
5. Optimize Resource Allocation: Allocate resources effectively to ensure task completion.
6. Identify Critical Path: Determine the critical path to identify tasks crucial for project duration.
7. Mitigate Risks: Incorporate buffers and contingency plans to address uncertainties.
8. Enable Monitoring and Control: Establish a framework for tracking progress against the schedule.
9. Facilitate Communication: Communicate the schedule and responsibilities to team members and stakeholders.
10. Ensure Project Success: Enhance the likelihood of project success through organized planning and execution.

**Project Schedules-**

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:

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

The first method in scheduling a software plan involves identifying all the functions required to complete the project. A good judgment of the intricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively. Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities. Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B, then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed. The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined. The end of every action is called a milestone. The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the milestones start getting delayed, then he has to handle the activities carefully so that the complete deadline can still be met.

**Sequencing and Scheduling Activities-**

Project and its activities must be clearly defined to achieve the target. An activity plan will contain the following factors:

* A project is basically, composed of number of interrelated activities.
* The initiation of a project happens only if atleast one activity is ready to start.
* An activity is clearly defined with its start and end point that produce good deliverables.
* Activity requiring resources must be analyzed well in advance and made available
* during the execution.
* Some activities would depend on other activities for them to complete.
* A project can attain its completion only when all activities have been completed.

Approaches to Identify Activities

* The various approaches used in identifying activities are:
* Activity-based approach
* Product-based approach
* Hybrid approach

Activity-based approach

* In the activity-based approach, all the activities are listed and created for the project.
* This is achieved by a brainstorming session where the entire project team analysis the
* various activities needed a0t different stages with the help of similar projects.
* This approach usually generates the list of activities using a work breakdown structure (WBS).
* WBS helps in identifying the lowest level of effort i.e. the task required to complete a project by breaking down into lower sets of tasks.
* Task defined at lower level includes everything that is required to complete the task at the higher level.
* The work breakdown structure provides an in-depth knowledge about the lowest level of activity that has to be completed.
* WBS is a refined structure that clearly defines the milestones that has to be achieved in accomplishing a specific task.
* The ordering of sequence of activities can also be done in this approach by defining those activities that have to be completed for others to start.
* In a purely activity-based approach, activities are identified and defined in fivelevels:
* Level 1 : Project – goals, objectives defined
* Level 2: Deliverables – software, manuals, training
* Level 3 : Components – work items, modules, tests
* Level 4 : Work-packages – major work items, related tasks
* Level 5 : Tasks – responsibility of an individual in accomplishing it

Product-based approach

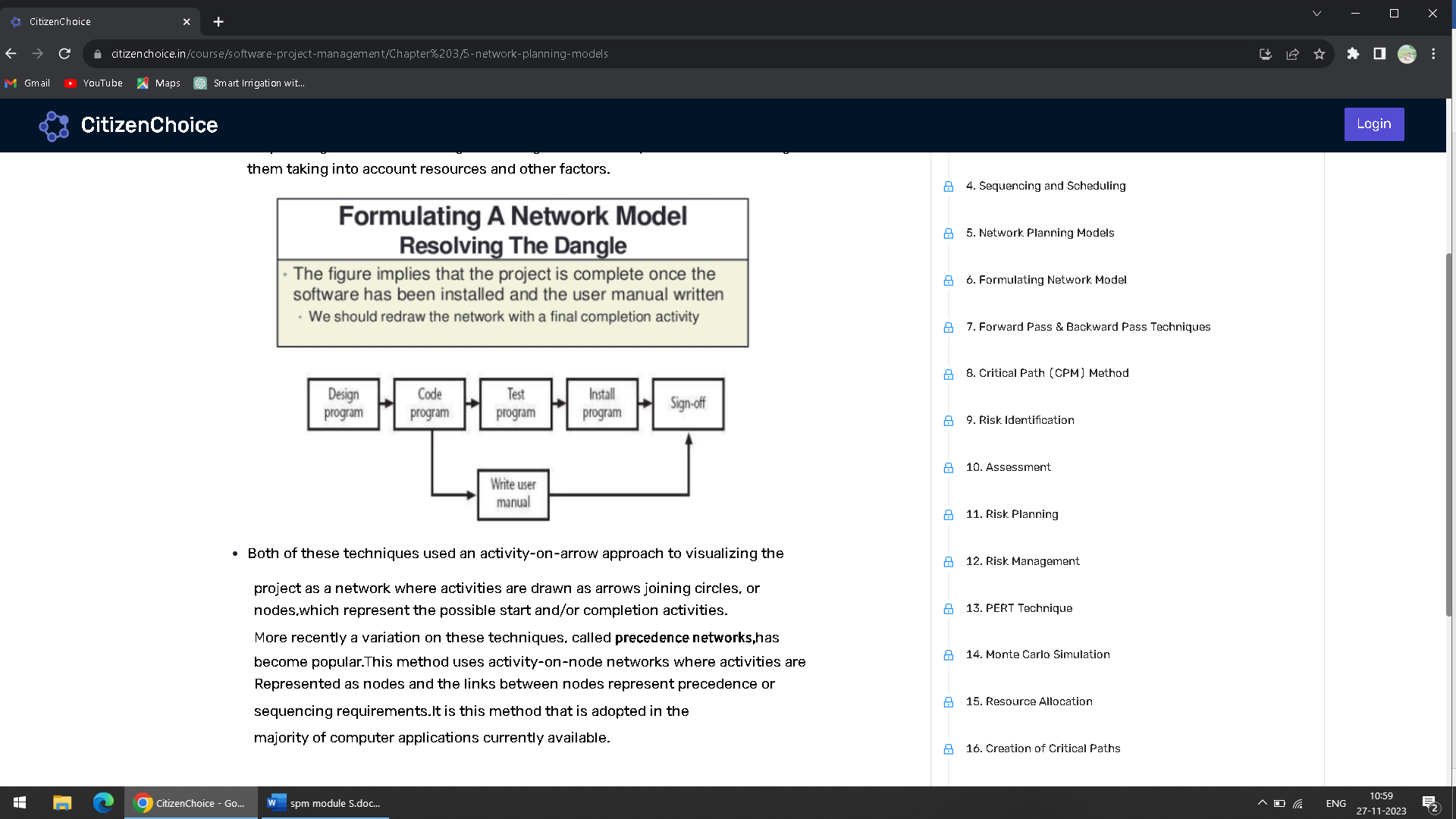
* The product-based approach produces a product breakdown structure along with a product flow diagram.
* The approach accepts the products as inputs which is transformed into an ordered list of activities.
* Product Flow Diagram do not leave out any activity from its ordered list and adopts a methodology which clearly specifies what are the products required and what are the activities required to produce the product.
* Using Structured Systems Analysis and Design Method (SSADM), a generic activity network can be derived for a project-specific product breakdown structure.
* The development of a PFD indicates the sequence of activities of the activity network.

Hybrid approach

* WBS deals with list of final deliverables whereas PBS deals in producing the products using the product flow diagram.
* Hybrid approach combines both the activity-based and product-based approach to structure both activities and products.
* Structuring of product-based or activity-based approach depend on the nature of the project type.

**Network Planning Models-**

* Scheduling  techniques model the project’s activities and their relationships as a “Network”. In network time flows from left to right.
* There are two best techniques:
  + - CPM(Critical Path Method)
    - PERT(Program Evaluation Review Technique)
* Both use the “Activity on Arrow” approach to visualize the project as a network where activities are drawn as arrows joining circles, or nodes which represent the possible start or completion of an activity or set of activities.
* Sequencing the tasks according to their logical relationship, and then scheduling them taking into account resources and other factors.
* Both of these techniques used an activity-on-arrow approach to visualizing the project as a network where activities are drawn as arrows joining circles, or nodes, which represent the possible start and/or completion activities. More recently a variation on these techniques, called **precedence networks, It** has become popular. This method uses activity-on-node networks where activities are Represented as nodes and the links between nodes represent precedence or sequencing requirements .It is this method that is adopted in the majority of computer applications currently available.

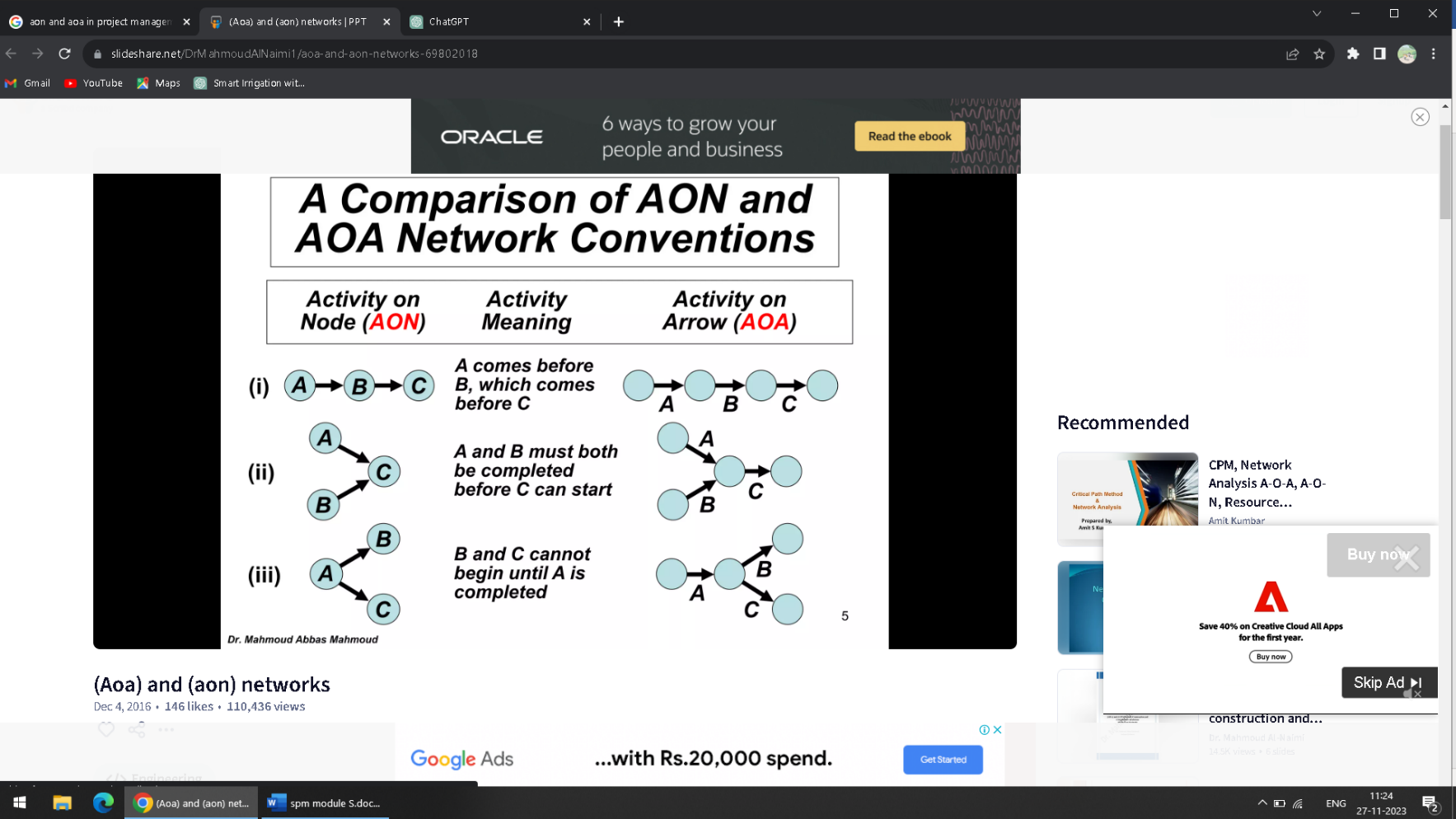


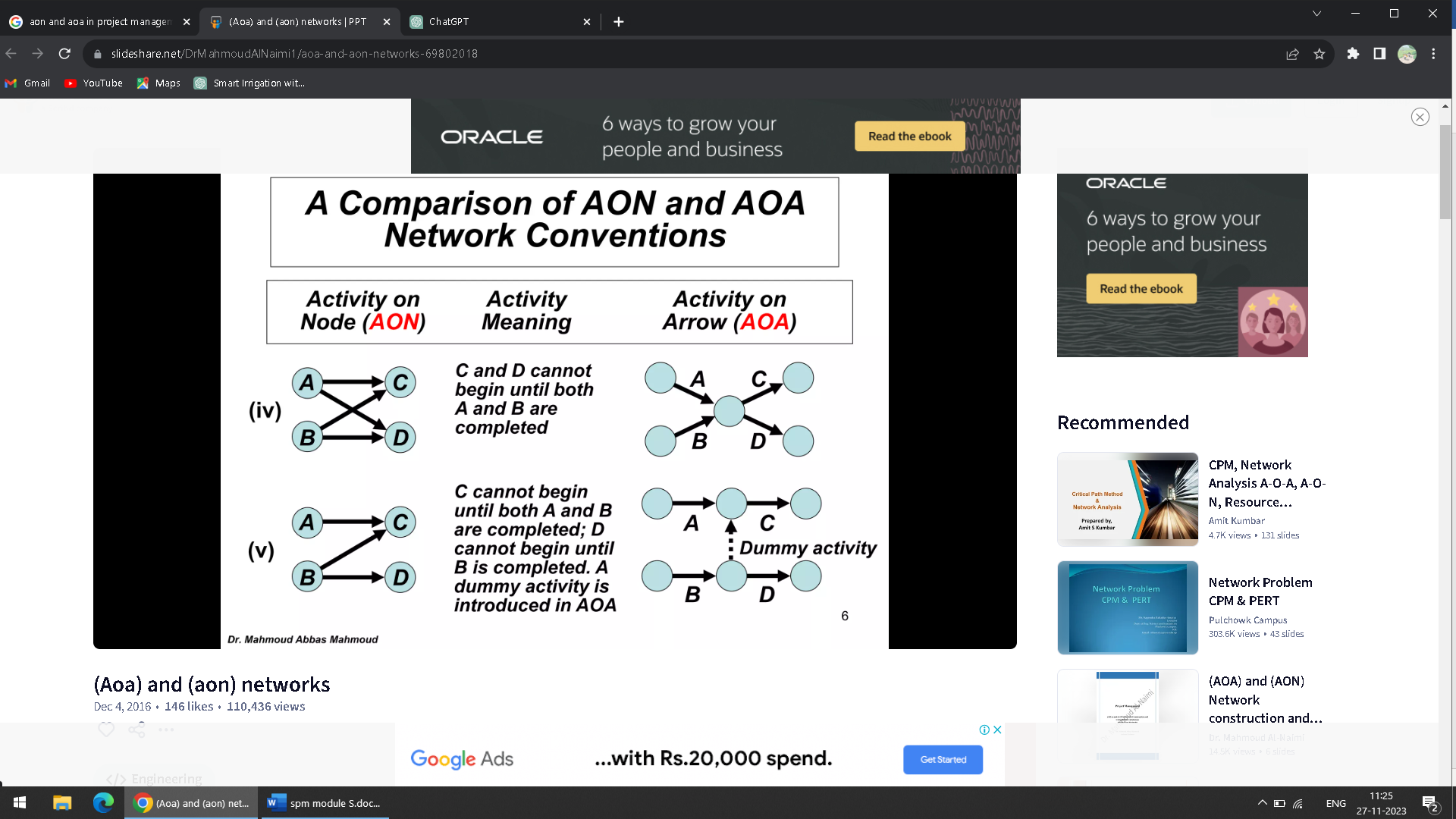
**AON and AOA-**

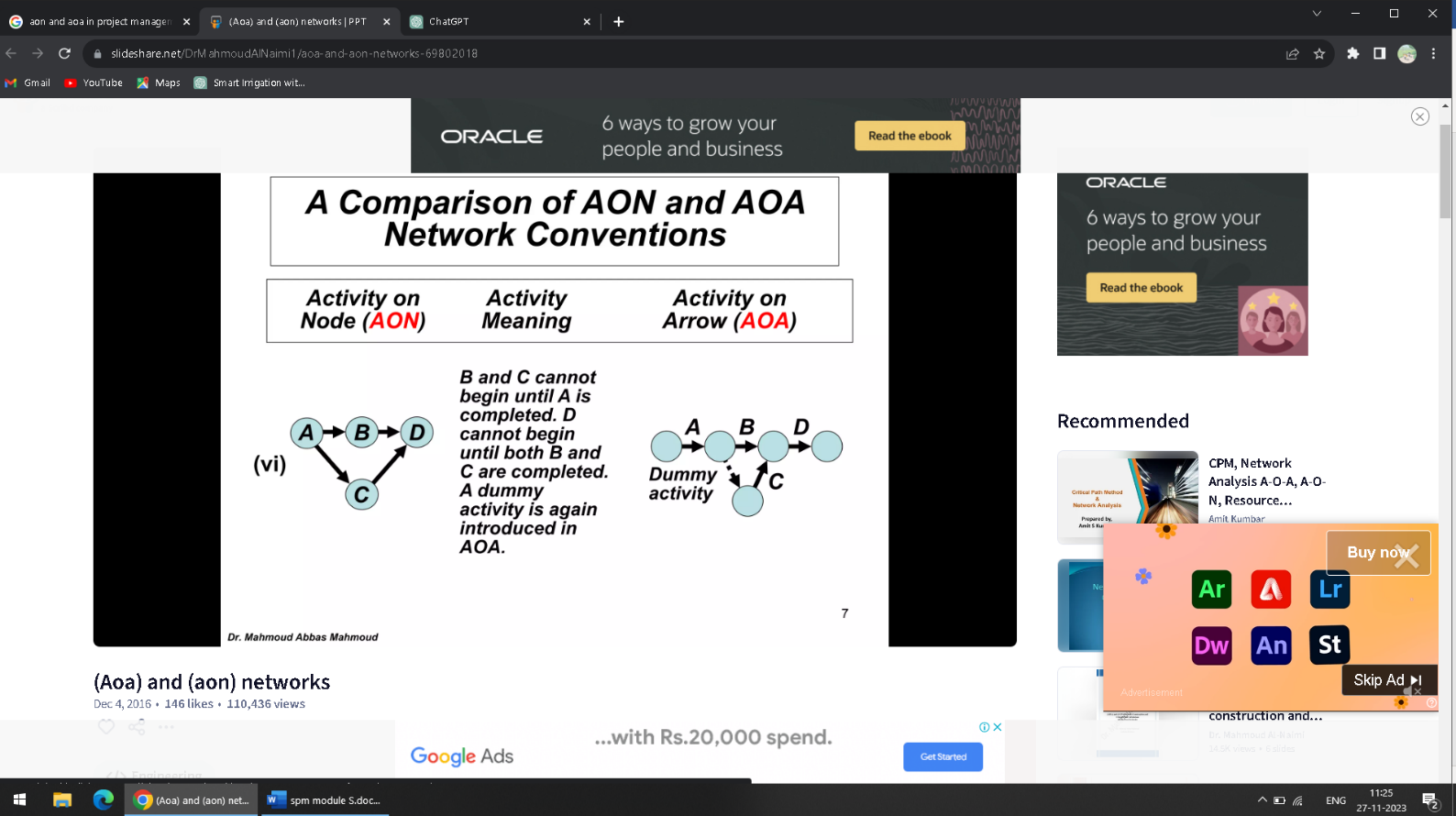
AON (Activity on Node) and AOA (Activity on Arrow) are two different network diagramming techniques used in project management to represent the flow of activities and their dependencies in a project. Both are part of the Critical Path Method (CPM), which is a widely used project management technique for scheduling and controlling project activities.

1. **Activity on Node (AON):**
   * In AON, the nodes of the network diagram represent the activities, and the arrows represent the dependencies between the activities.
   * The activities are represented by rectangles or nodes, and arrows connecting the nodes indicate the precedence relationships or dependencies between the activities.
   * The advantage of AON is that it is visually straightforward and easy to understand.
2. **Activity on Arrow (AOA):**
   * In AOA, the arrows in the network diagram represent the activities, and the nodes represent events or milestones.
   * The events are represented by circles or nodes, and the arrows connecting the events represent the activities. The length of the arrow indicates the duration of the activity.
   * AOA is less commonly used than AON, but it was historically used in some project management approaches.

In both AON and AOA, the critical path can be determined to identify the longest path through the project, indicating the minimum time required for project completion. The critical path is essential for project managers to focus on to ensure the project is completed within the specified time frame. While AON is more prevalent in modern project management, AOA is considered a historical approach and is not as commonly used today.







**Identifying critical activities-**

Identifying critical activities in software project management is crucial for ensuring the successful completion of a project. Critical activities are those tasks or processes that, if delayed or not executed properly, could have a significant impact on the project timeline, cost, or overall success. Here are some key steps to identify critical activities in software project management:

* Work Breakdown Structure (WBS): Break down the entire project into smaller, manageable tasks. Create a Work Breakdown Structure (WBS) that organizes these tasks hierarchically. This will help you visualize the project's scope and identify dependencies between tasks.
* Dependency Analysis: Identify dependencies between tasks. Dependencies determine the order in which tasks need to be completed. Tasks with dependencies are often critical because delays in one task can affect the entire project timeline.
* Critical Path Method (CPM): Use the Critical Path Method to identify the critical path through the project network. The critical path is the longest sequence of dependent tasks that determines the project's overall duration. Tasks on the critical path are inherently critical to the project's timeline.
* Resource Constraints: Identify tasks that are constrained by resource availability, such as specialized skills or equipment. If these resources are limited, the associated tasks become critical as delays in resource allocation can impact the project.
* Risk Assessment: Evaluate potential risks that could impact the project. Tasks associated with high-risk elements are often critical. Mitigating risks and addressing critical tasks early in the project can prevent larger issues later on.
* Stakeholder Input: Consult with project stakeholders, including team members, clients, and end-users, to identify tasks that are critical to meeting their expectations. Understanding the priorities and expectations of stakeholders is essential for project success.
* Milestone Identification: Identify key milestones in the project timeline. Milestones are points in time that mark the completion of a significant phase or deliverable. Tasks leading to these milestones are often critical.
* Buffer Analysis: Assess the presence of buffers or slack in the project schedule. Tasks with little or no slack are typically critical, as any delay in these tasks will directly impact the overall project timeline.
* Historical Data: Review past projects and their timelines to identify common critical activities. Lessons learned from previous projects can help in anticipating and managing critical tasks in the current project.
* Continuous Monitoring: Regularly monitor the project's progress and reassess critical activities as the project evolves. Changes in scope, resources, or external factors may affect the critical path.
* By employing these techniques, project managers can effectively identify and prioritize critical activities, allowing them to allocate resources, manage risks, and ensure the successful completion of the software project.

**Schedule Compression**

Schedule compression offers two important techniques for scheduling time so projects are executed on schedule. Understanding these techniques and their application in compressing schedules can be very useful, whether in professional or personal spheres.

Schedule compression refers to techniques used when a project manager wants to shorten the duration of the project without changing the scope of the project. It can be used when a project falls behind schedule and needs to catch up or to finish the project sooner than originally scheduled.

The two techniques you can use to shorten the project duration while maintaining the project scope are

* fast tracking
* crashing.

**Fast Tracking**

Fast-tracking in project management refers to the practice of concurrently completing tasks that would normally be completed sequentially. Projects that are being fast-tracked may indicate a number of factors, including the possibility that the project will not be completed on schedule.

Fast-tracking is a technique where activities that would have been performed sequentially using the original schedule are performed in parallel. In other words, fast tracking a project means the activities are worked on simultaneously instead of waiting for each piece to be completed separately. But fast tracking can only be applied if the activities in question can actually be overlapped.

When you need to compress a schedule, you should consider this technique first, because fast tracking usually does not involve any costs. This technique simply rearranges the activities in the original schedule.

Although fast tracking may not result in an increase in the cost, it leads to an increase in the risk, because activities now being performed in parallel may lead to needing to rework or rearrange the project. Reworking the project can also waste even more time.

**Crashing**

Crashing is the technique to use when fast tracking has not saved enough time on the project schedule. With this technique, resources are added to the project for the least cost possible. Cost and schedule tradeoffs are analyzed to determine how to obtain the greatest amount of compression for the least incremental cost. And crashing is expensive because more resources are added to the project.

Crashing analyzes and categorizes activities based on the lowest crash cost per unit time, allowing the team working the project to identify the activities that will be able to deliver the most value at the least incremental cost. The results of a crash analysis are usually presented in a crash graph, where activities with the flattest slope are the ones that will be considered first—they lead to an equal amount of time savings, but have a smaller increase in cost. Crashing only works if the additional resources will actually achieve completing the project sooner.

When the crashing approach is used, any additional costs associated with rushing the project are reviewed against the possible benefits of completing the project within a shorter time span. In addition, you should consider other items when performing a crash analysis, including adding more resources to the project, allowing additional overtime, and paying extra to receive delivery of critical components more quickly, among others.

**What is Risk?**

"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.

We need to differentiate risks, as potential issues, from the current problems of the project. Different methods are required to address these two kinds of issues. 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 three main classifications of risks which can affect a software project:

1. Project risks
2. Technical risks
3. Business risks

**1. Project risks:** Project risks concern differ forms of budgetary, schedule, personnel, resource, and customer-related problems. A vital project risk is schedule slippage. Since the software is intangible, it is very tough to monitor and control a software project. It is very tough to control something which cannot be identified. For any manufacturing program, such as the manufacturing of cars, the plan executive can recognize the product taking shape.

**2.Technical risks:** Technical risks concern potential method, implementation, interfacing, testing, and maintenance issue. It also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty, and technical obsolescence. Most technical risks appear due to the development team's insufficient knowledge about the project.

**3. Business risks:** This type of risks contain risks of building an excellent product that no one need, losing budgetary or personnel commitments, etc.

**Other risk categories**

**1. Known risks:** Those risks that can be uncovered after careful assessment of the project program, the business and technical environment in which the plan is being developed, and more reliable data sources (e.g., unrealistic delivery date)

**2. Predictable risks:** Those risks that are hypothesized from previous project experience (e.g., past turnover)

**3. Unpredictable risks:** Those risks that can and do occur, but are extremely tough to identify in advance.

**Principle of Risk Management**

1. **Global Perspective:** In this, we review the bigger system description, design, and implementation. We look at the chance and the impact the risk is going to have.
2. **Take a forward-looking view:** Consider the threat which may appear in the future and create future plans for directing the next events.
3. **Open Communication:** This is to allow the free flow of communications between the client and the team members so that they have certainty about the risks.
4. **Integrated management:** In this method risk management is made an integral part of project management.
5. **Continuous process:** In this phase, the risks are tracked continuously throughout the risk management paradigm.

**Risk Management Activities**

**Proactive Risk Management**

Proactive risk management involves taking preventive measures to address potential risks before they occur. It begins with a thorough risk assessment process, where the project team identifies and analyses potential risks based on past experiences, industry knowledge, and project-specific factors.

The goal is anticipating and understanding the risks that could impact the project’s success. Once risks are identified, strategies and actions are developed to mitigate or eliminate them. This may include conducting feasibility studies, performing comprehensive system analyses, employing robust testing methodologies, or implementing redundant systems. The focus is on proactive planning and preparation to minimize the likelihood and impact of potential risks.

**Reactive Risk Management**

Reactive risk management is employed when risks have already occurred or are currently affecting the project. It involves responding to and addressing risks in a timely manner to minimize their impact on the project’s objectives. This approach requires a rapid and efficient response to mitigate the effects of risks and prevent further negative consequences.

When a risk is identified, the project team initiates contingency plans and implements corrective actions. This may involve reallocating resources, adjusting project schedules, conducting additional testing, or seeking expert advice. The goal is to react quickly and effectively to minimize the impact of risks and keep the project on track.

**Technical Risk Management**

Technical risk management focuses on addressing risks related to the technical aspects of software development. It involves identifying potential technical challenges that could impact the project’s success. This may include risks associated with software complexity, integration issues, performance limitations, or security vulnerabilities.

To manage technical risks, the project team conducts thorough assessments to identify potential problems. Strategies are then implemented to mitigate these risks, such as using proven development methodologies, performing rigorous testing and quality assurance, or allocating sufficient time and resources for system integration. The aim is to ensure the technical aspects of the software are robust, reliable, and meet the project’s requirements.

**Project Risk Management**

Project risk management involves addressing risks that arise from project management and execution. These risks can impact project schedules, budgets, resource allocation, and overall project success. Project-related risks include unrealistic deadlines, inadequate resource planning, poor communication, or scope creep.

To manage project risks, the project team identifies potential risks early on and develops strategies to minimize their impact. This may involve setting realistic project timelines, conducting regular progress reviews, ensuring effective communication among team members and stakeholders, and implementing project management best practices. The focus is on proper planning, monitoring, and control to deliver the project successfully.

**Organizational Risk Management**

Organizational risk management focuses on risks that arise from within the organization or project environment. These risks can stem from inadequate stakeholder engagement, insufficient support from management, lack of expertise, or ineffective collaboration among team members.

To manage organizational risks, the project team works on creating an environment conducive to project success. This may involve clear communication channels, ensuring stakeholder involvement and support, providing adequate training and resources, and encourage a positive team culture. By addressing organizational risks, the project team can enhance coordination, collaboration, and overall project performance.

**Business Risk Management**

Business risk management in software engineering focuses on risks related to the business context in which the software project operates. These risks may include market competition, changing customer demands, financial constraints, or regulatory compliance requirements.

To manage business risks, the project team assesses the external factors that could impact the project’s success. Strategies are then developed to address these risks, such as conducting market research, adapting to evolving customer needs, ensuring financial viability, or complying with relevant regulations and standards. The aim is to align the software solution with the business objectives and mitigate potential business-related risks.

**Continuous Risk Management**

Continuous risk management emphasizes the iterative and ongoing nature of risk management throughout the software development lifecycle. It recognizes that risks can evolve and emerge at different project stages and requires a proactive approach to manage them effectively.

This approach involves regularly reviewing and updating the risk management plan, identifying new risks as they arise, and reassessing existing risks. It also includes monitoring the status of identified risks, implementing control measures, and learning from past experiences to improve risk management practices. By adopting continuous risk management, the project team can respond to emerging risks promptly and ensure the project’s overall success.

**What is a PERT Chart?**

PERT or the Program Evaluation and Review Technique is a method that analyzes the time required to complete each task and its associated dependencies, and to determine the minimum amount of time required to complete a certain project. The process takes into consideration three different time estimates:

* Optimistic Time (To): The minimum amount of time required to complete the project, assuming everything goes better than expected.
* Pessimistic Time (Tp): The maximum time required to complete the task, assuming things go wrong.
* Most Likely Time (Tm): The most likely amount of time required to complete the tasks, assuming everything goes alright.

**PERT Chart vs Gantt Chart**

PERT (Program Evaluation and Review Technique) and Gantt charts are both tools used in project management, but they serve different purposes:

**PERT Charts:**

* Visualize task dependencies and their interrelatedness in a project
* Emphasize the sequence and timing of tasks
* Are helpful in planning and scheduling projects with multiple tasks and dependencies

**Gantt Charts:**

* Visualize task progress over time
* Emphasize the duration of tasks and their start/end dates
* They are useful for monitoring progress and ensuring tasks are completed on time.

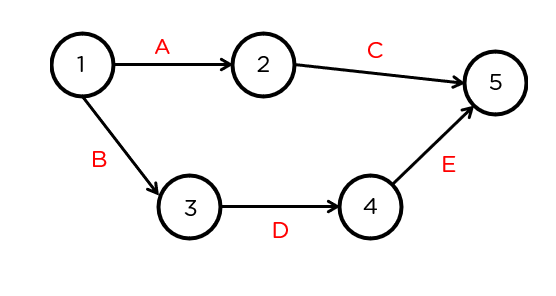
In summary, PERT charts focus on the relationships between tasks, while Gantt charts focus on the progress of jobs over time.

**How Do You Make a PERT Chart?**

* Define the project scope: Determine the objectives and goals of the project, and list all the tasks required to achieve those objectives.
* Establish task dependencies: Identify the dependencies between tasks and determine the order in which tasks must be completed.
* Determine task duration: Estimate the time each task will take to complete.
* Create a network diagram: Use arrows to connect the tasks and show their dependencies on each other. Number the tasks and events, and list their estimated duration.
* Add critical path information: Determine the critical path and the sequence of tasks that determines the minimum overall project duration.
* Update the chart regularly: Revisit the PERT chart regularly to reflect changes in the project, such as changes in task dependencies, duration, or priority.
* Present the chart: The final PERT chart should clearly show the relationships between tasks, the critical path, and the estimated duration of each task.

**PERT Chart Terms and Concepts**

Before we get into the [PERT Analysis](https://www.simplilearn.com/pert-and-cpm-important-tools-of-project-management-rar225-article) process, we must talk about some important concepts: Events and Activities. Let’s understand these terms with the help of a network diagram (which is the final output of the method).



**Event**

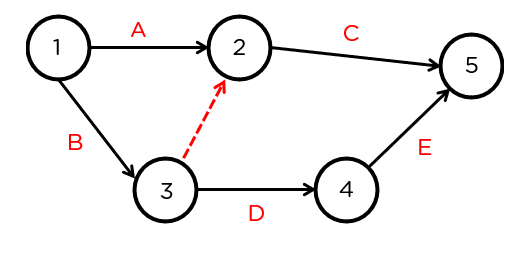
A circle represents events and will occur at the start and end of an activity. Event 1 is the tail event, and Event 2 is the head event. In the case of our example, node 1 will be referred to as the tail event, and 2 will be referred to as the head event.

**Activity**

Activities represent action and consumption of resources like time, money, and energy required to complete the project. In the case of our example, A, B, C, D, and E represent the activities taking place between their respective events.

**Dummy Activity**

A dummy activity represents a relationship between two events. In the case of the example below this, the dotted line represents a relationship between nodes 3 and 2.  
The activity between these nodes will not have any value.



Other rules that need to be considered are:

* The network should have a unique starting and ending node.
* No activity can be represented by more than a single arc (the line with an arrow connecting the events) in the network.
* No two activities can have the same starting and ending node.

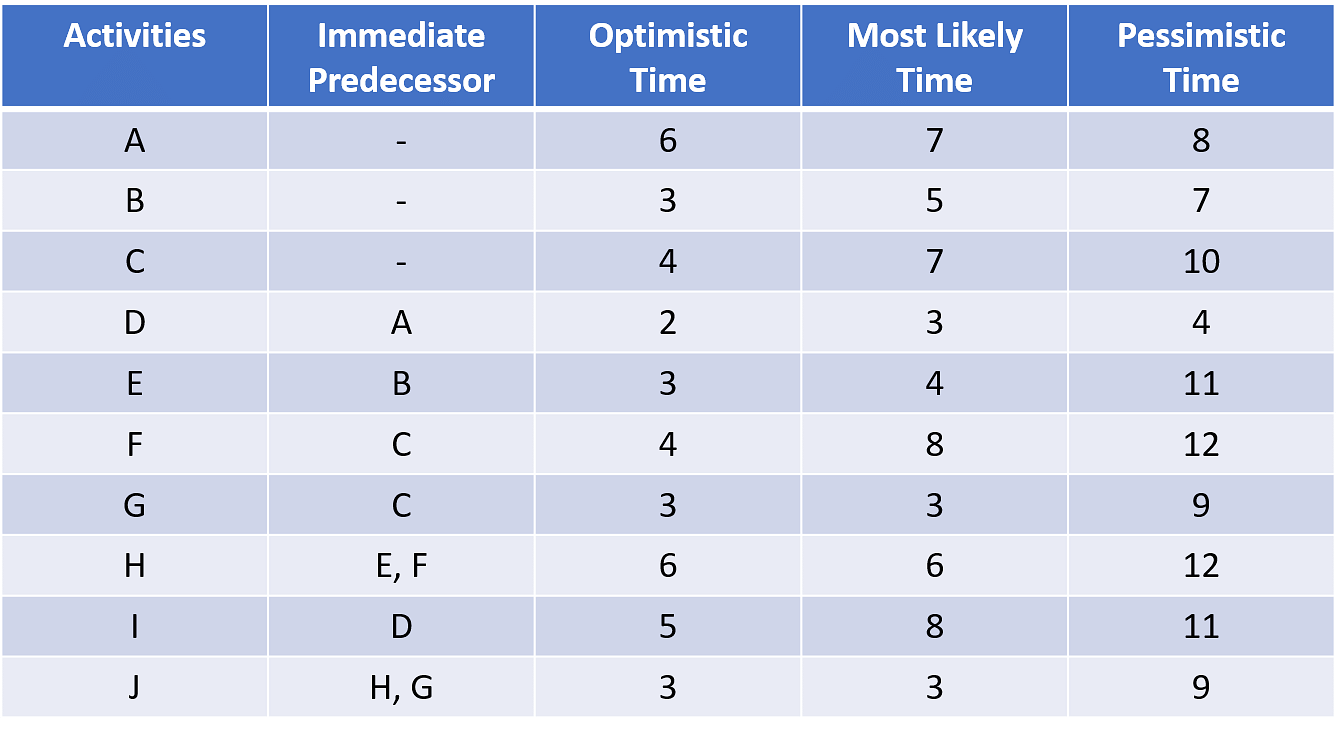
**The PERT Analysis Method**

In the question here, we have three objectives:

1. Draw the network diagram.

2. Find the mean and variance.

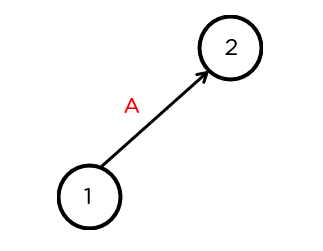
3. Find the critical path and estimated time of completion.



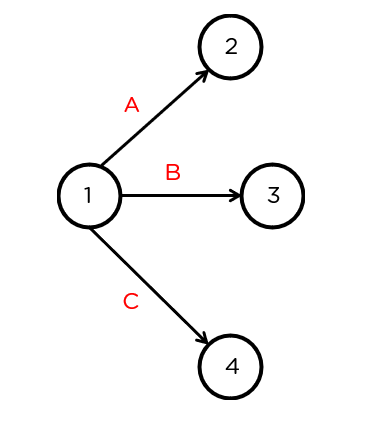
Now, let’s draw the network diagram.

First, let’s look at the activities and their immediate predecessors.

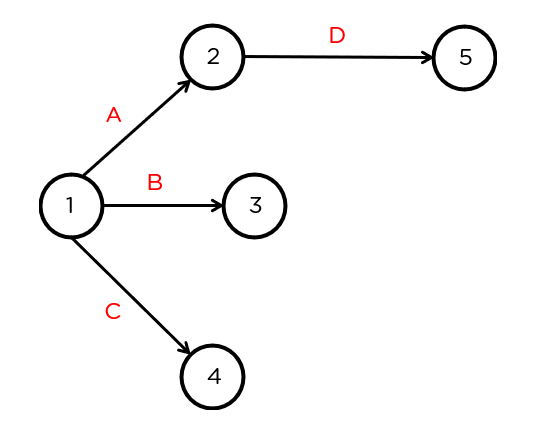
We can see that activities A, B, and C don’t have any immediate predecessors. This means that we can draw individual arcs to each of them. Let’s draw the nodes for the first activity, activity A. We can see that activity A acts as the immediate predecessor for the activity D.



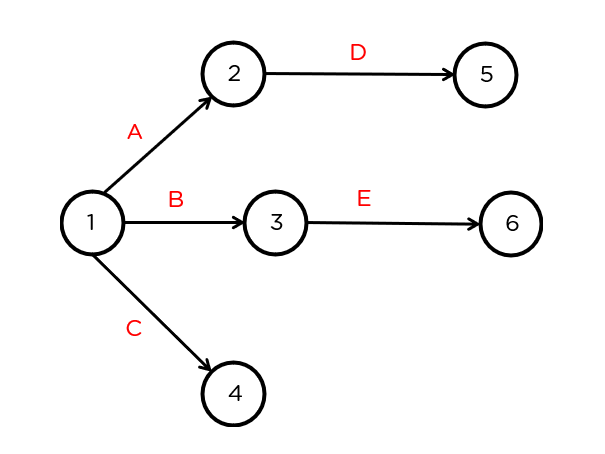
Similarly, activities B and C don’t have any immediate predecessors and hence, can be directly connected to node 1. Node B acts as an immediate predecessor for E, while node C acts as the immediate predecessor for activities F and G. Let’s go ahead, and draw that.



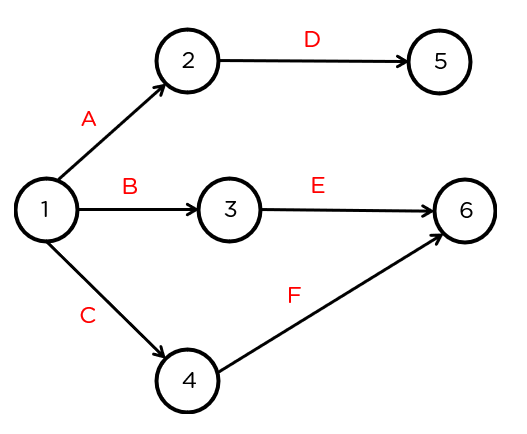
Let’s have a look at activity D. This activity is the immediate predecessor for activity A. This means that we can directly draw an arc from node 2.



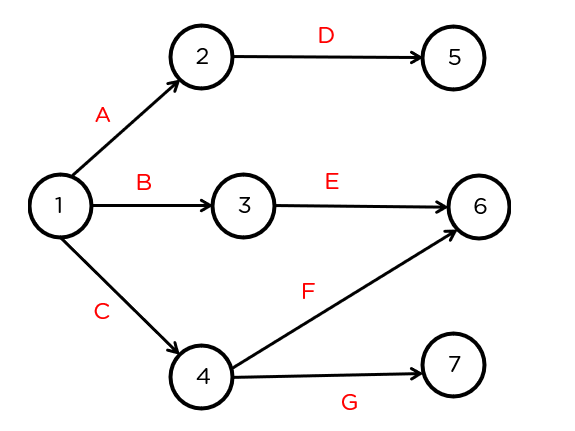
Now, we’ve drawn activities A, B, C, and D as part of the PERT analysis. Now, looking at activity E, it acts as the immediate predecessor to activity H along with activity F. Since it’s preceded only by activity B, we can directly connect it to node 3.



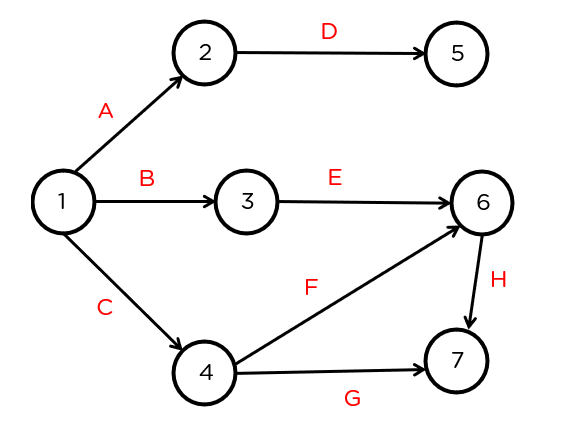
Now, for activity F. If we have a look at the table, we can see that a combination of the activities E and F act as immediate predecessors for activity H. This means that activities E and F need to come together at the node 6.



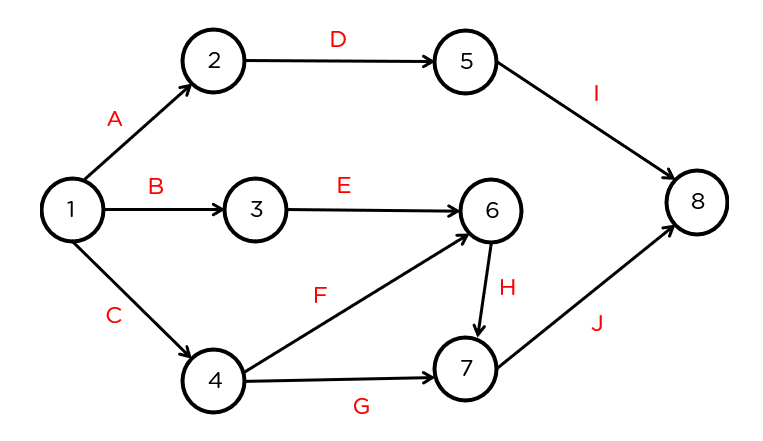
Next up, let’s have a look at activity G. It is immediately preceded by activity C, and acts as an immediate predecessor for activity J, along with activity H. Since it’s an independent activity, we can draw it like so:



For activity H, we can see that it and G act as immediate predecessors for activity J. This means that nodes 6 and 7 need to be connected.

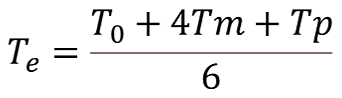


And finally, we activities I and J. These activities don’t act as immediate predecessors for any other activity. This means that they’ll connect directly to the final node.

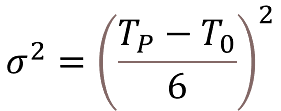


Now that we’ve created the network diagram, let’s move ahead. Next, as part of the PERT analysis, let’s have a look at how to determine the mean and variance.

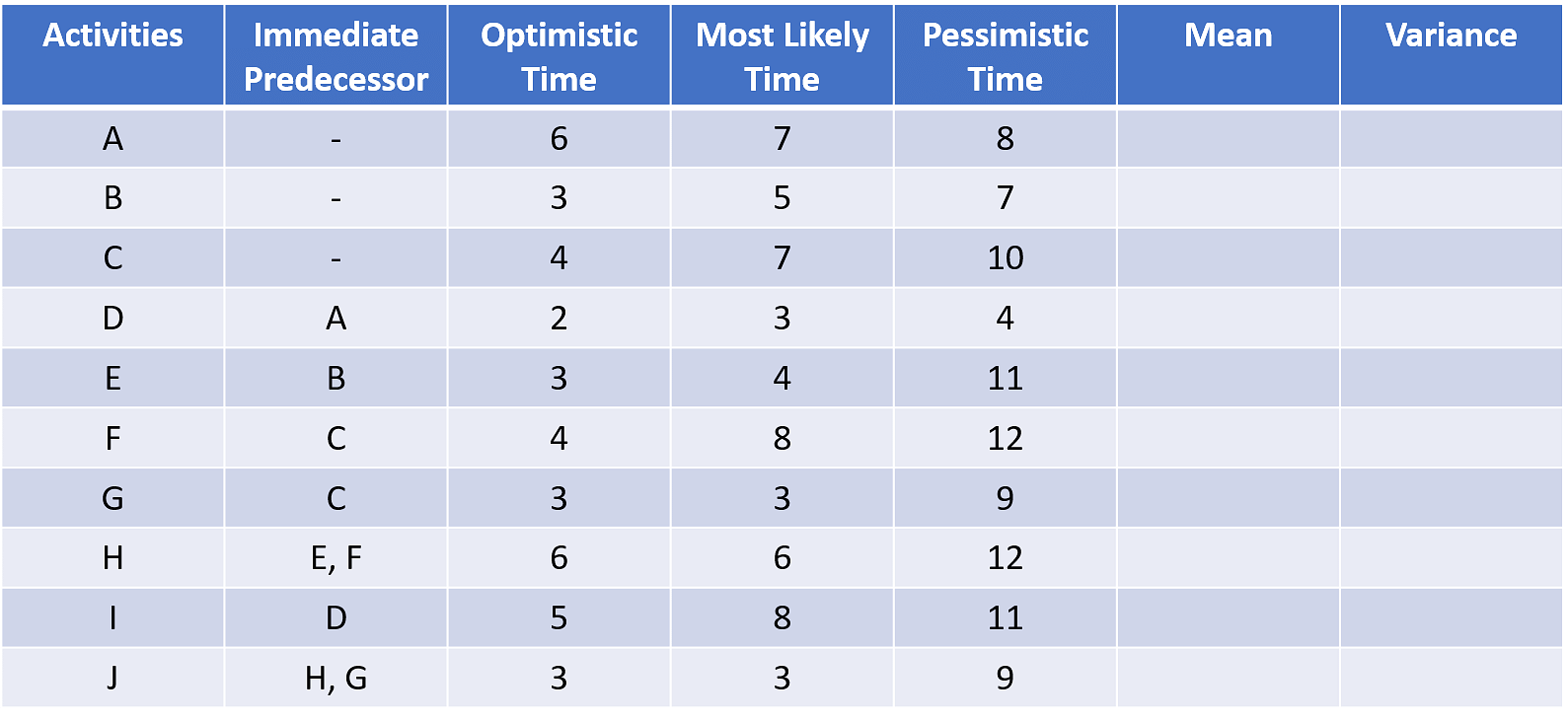
The mean, which is also the estimated time can be determined using the formula:



We can calculate the variance using this formula:



Let’s apply the formula to each activity.



For activity A,

The mean will be: (To + 4\*Tm + Tp) /6 =  (6 + 4\*7 + 8) /6 = 7

For activity B,

The mean will be: : (To + 4\*Tm + Tp) /6 = (3 + 4\*5 + 7) /6 = 5

For activity C,

The mean will be: : (To + 4\*Tm + Tp) /6 = (4 +4\*7 +10) /6 = 7

For activity D,

The mean will be: : (To + 4\*Tm + Tp) /6 = (2 + 4\*3 +4) /6 = 3

For activity E,

The mean will be: : (To + 4\*Tm + Tp) /6 = (3 + 4\*4 + 11) /6 = 5

For activity F,

The mean will be: : (To + 4\*Tm + Tp) /6 = (4 + 4\*8 + 12) /6 = 8

For activity G,

The mean will be: : (To + 4\*Tm + Tp) /6 = (3 + 4\*3 + 9) /6 = 4

For activity H,

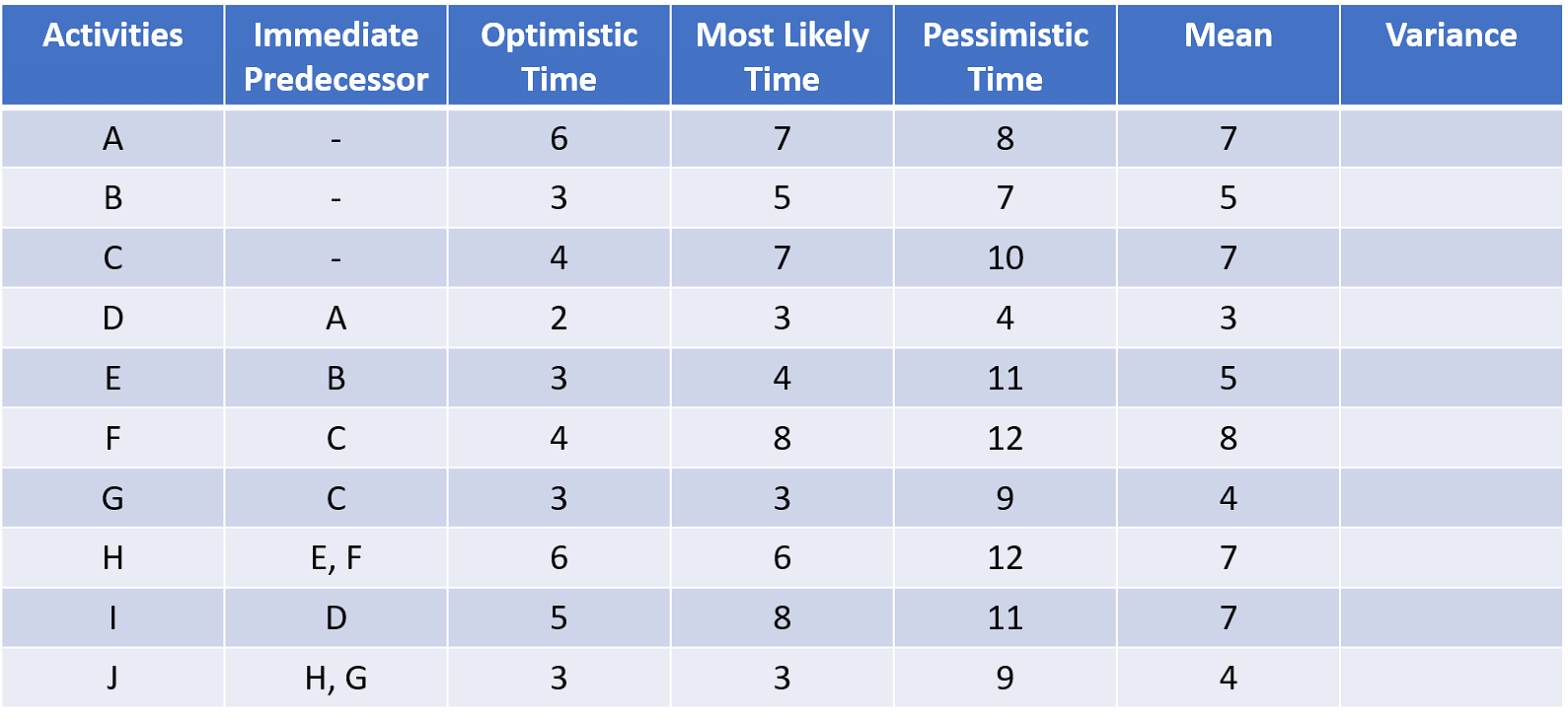
The mean will be: : (To + 4\*Tm + Tp) /6 = (6 + 4\*6 + 12) /6 = 7

For activity I,

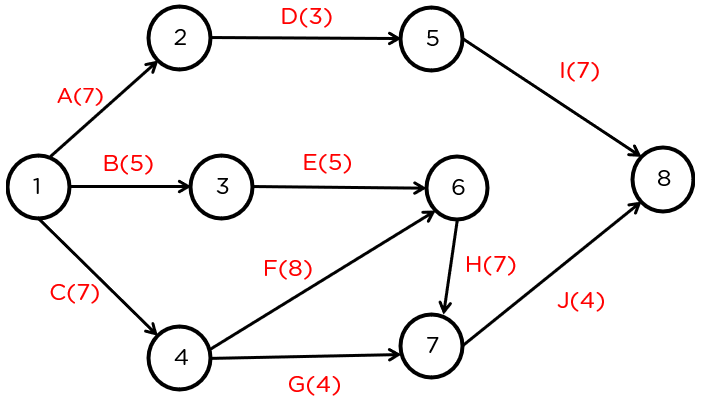
The mean will be: : (To + 4\*Tm + Tp) /6 = (5 + 4\*8 + 11) /6 = 7

For activity J,

The mean will be: : (To + 4\*Tm + Tp) /6 = (3 + 4\*3 + 9) /6 = 4



This mean can be applied to the network, to each of the activities.



Now, let’s find the variance for each of these activities.

2 = [(Tp - To) /6]2

For activity A:

2 = [(Tp - To) /6]2= 2 = [(8 - 6) /6]2= 0.11

For activity B:

2 = [(Tp - To) /6]2= 2 = [(7 - 3) /6]2= 0.44

For activity C:

2 = [(Tp - To) /6]2= 2 = [(10- 4) /6]2= 1

For activity D:

2 = [(Tp - To) /6]2= 2 = [(4 - 2) /6]2= 0.11

For activity E:

2 = [(Tp - To) /6]2= 2 = [(11 - 3) /6]2= 1.77

For activity F:

2 = [(Tp - To) /6]2= 2 = [(12 - 4) /6]2= 1.77

For activity G:

2 = [(Tp - To) /6]2= 2 = [(9 - 3) /6]2= 1

For activity H:

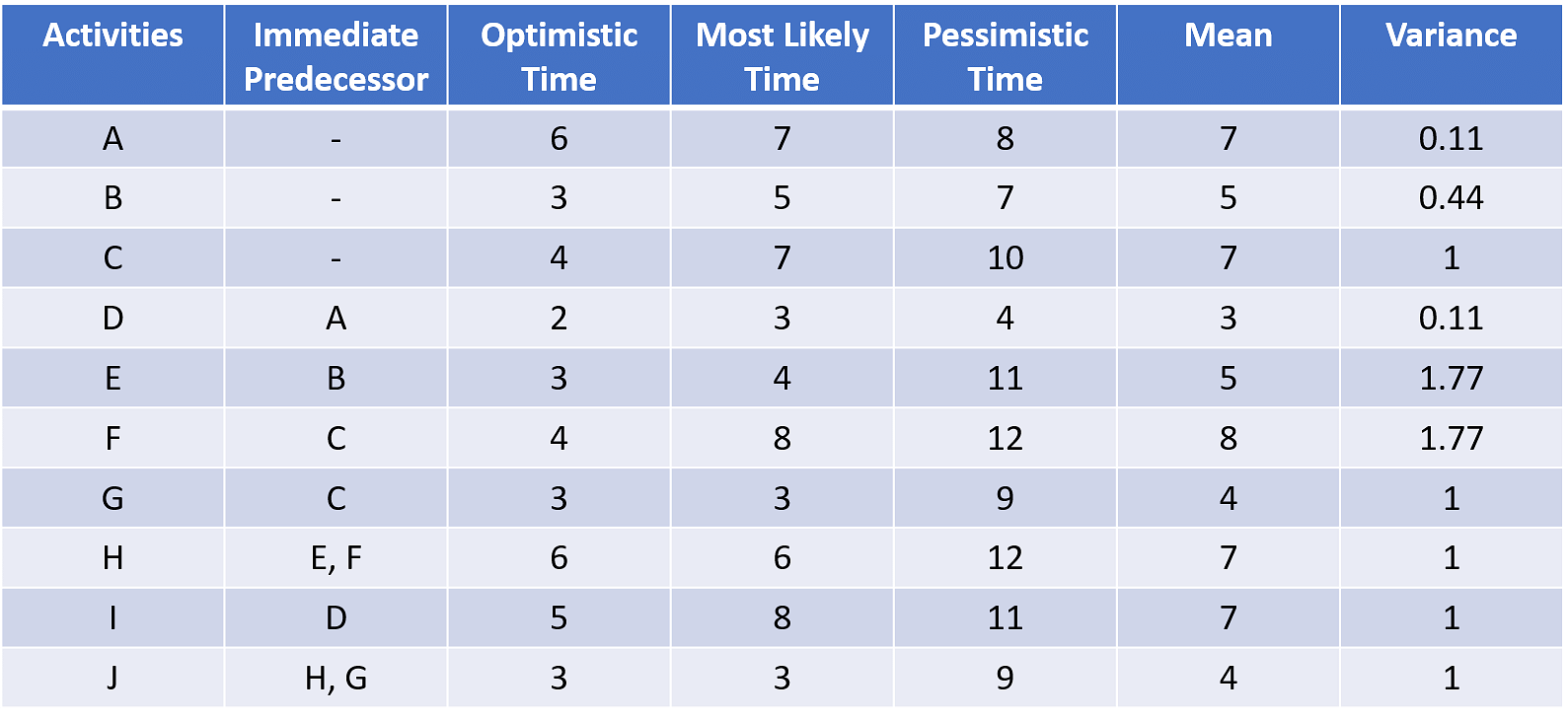
2 = [(Tp - To) /6]2= 2 = [(12 - 6) /6]2= 1

For activity I:

2 = [(Tp - To) /6]2= 2 = [(11 - 5) /6]2= 1

For activity J:

2 = [(Tp - To) /6]2= 2 = [(9 - 3) /6]2= 1



Now, for the third part of the PERT analysis. We need to find the critical path and the estimated time.

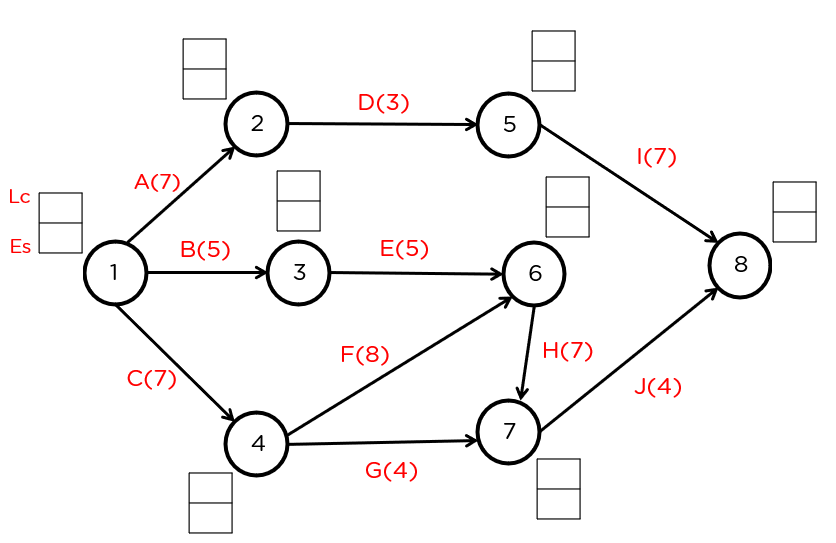
For this, we’ll need to find two values, Earliest Start Time (Es) and Latest Completion Time (Lc).

The process of determining the Es for all events is called a forward pass.

The process of determining the Lc for all events is called a backward pass.

Let’s get into the forward pass. For this first, we must create boxes at all nodes. We then divide these into two. The lower half of the box represents the earliest start time of the node, while the lower half represents the latest completion time.

Your network diagram should look something like this.



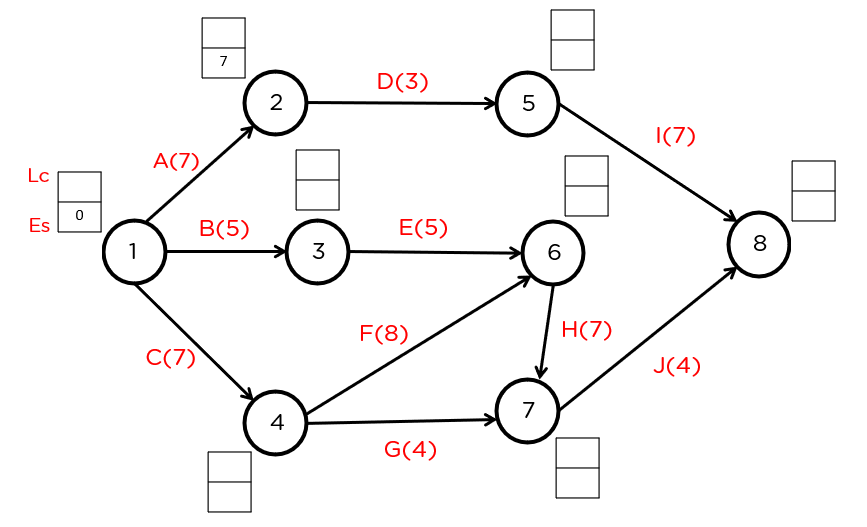
For this, we’ll be using the formula, Esj = max (Esi + Dij)

Which when simplified, the earliest start time for the second node (head node), is the maximum of the combination of the earliest start time of the tail node and the duration between the two nodes.

So, for node 1, the earliest start time is always zero.

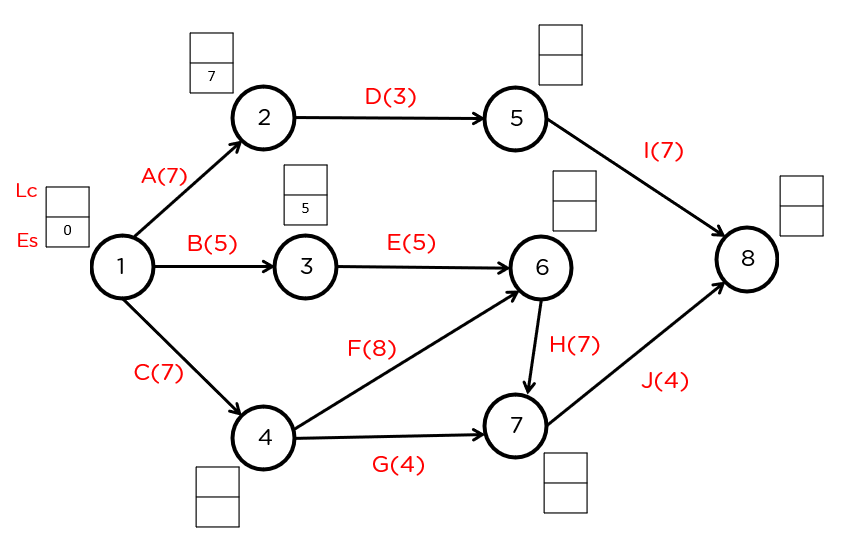
For node 2,

Es2 = 0 (Es1) + 7(D1-2) = 7



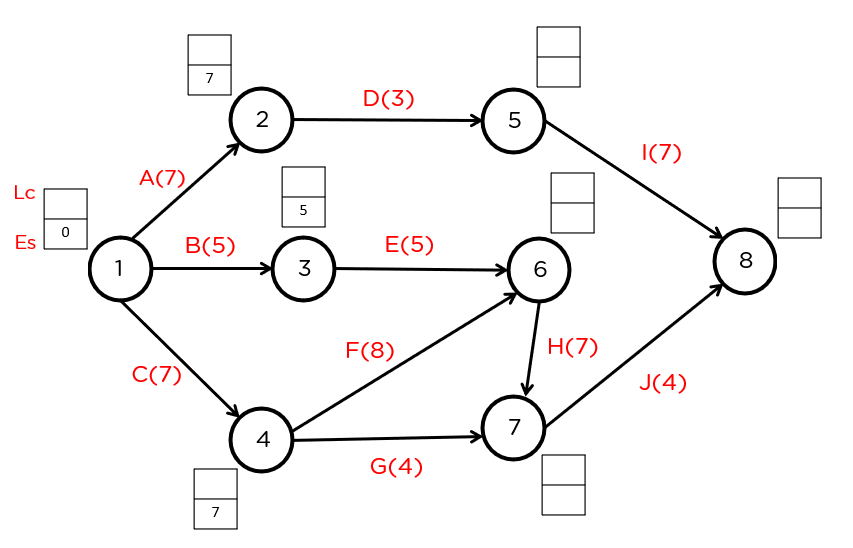
Next node 3.

Es3 = 0(Es1) + 5(D1-3) = 5



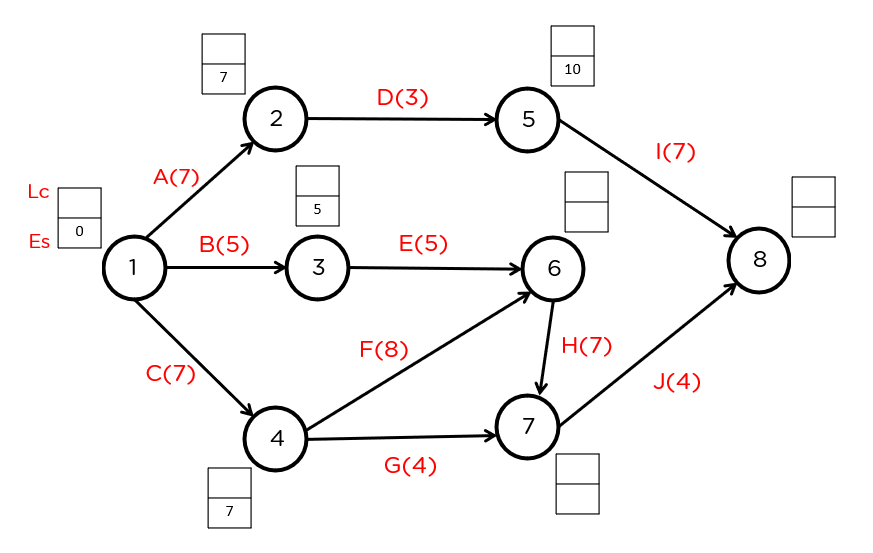
Now, for node 4.

Es4 = 0(Es1) + 7(D1-4) = 7



Next, we have node 5.

Es5 = 7(Es2) + 3(D2-5) =10



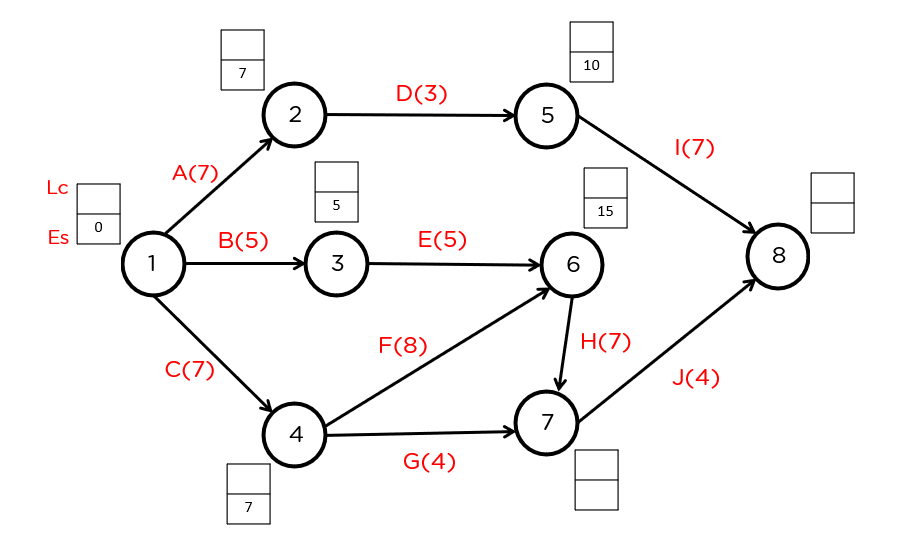
Now for node 6.

Since there are two arcs connecting to the node, we need to choose the maximum of the two options available.

Es6 = 5(Es3) + 5(D3-6) = 10 or

Es6 = 7(Es4) + 8(D4-6) = 15

We must choose the maximum of the two, so we’ll select 15.

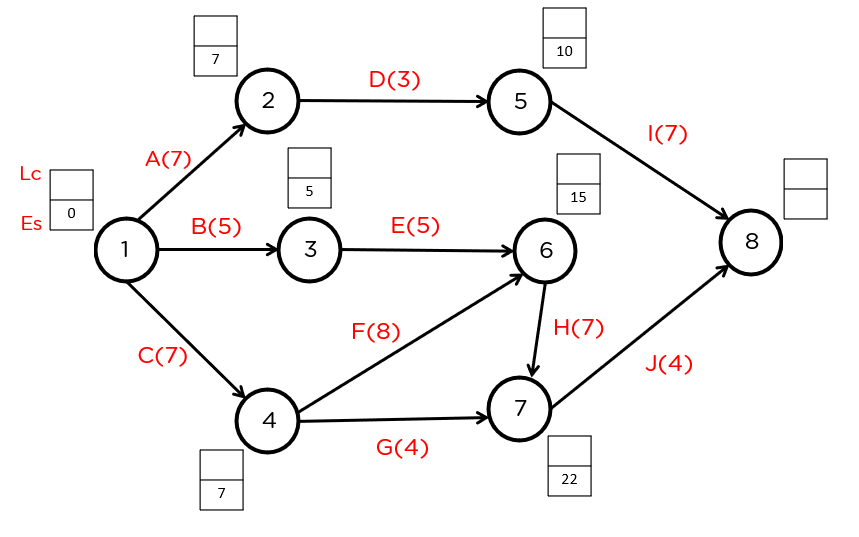


Next, we have node 7. Since there are two nodes connecting to it; we need to choose the maximum among the two options.

Es7 = 15(Es6) + 7((D6-7) = 22 or

Es7 = 7(Es4) + 4(D4-7) = 11

We’ll need to choose the maximum, and we’ll choose 22.

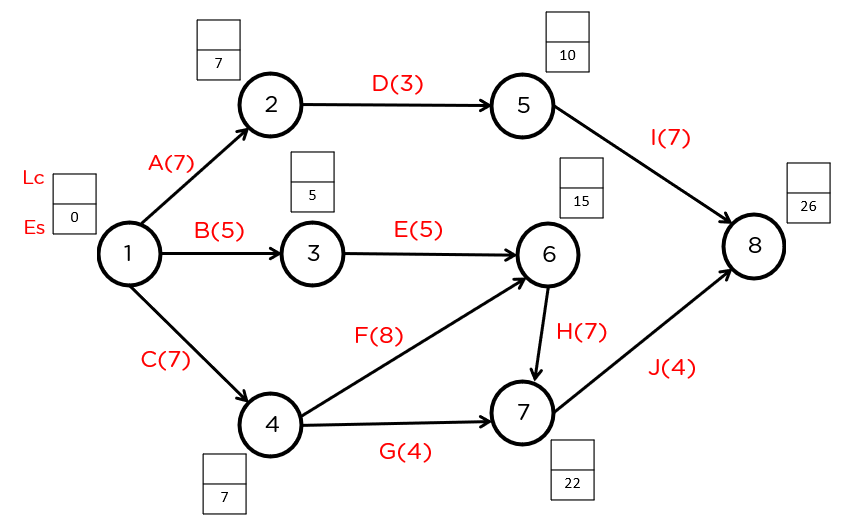


And finally, we’ll need to find the earliest start time for node 8.

Es8 = 10(Es5) + 7(D5-8) = 17 or

Es8 = 22(Es7) + 4(D7-8) = 26

Since we need to choose the maximum value, we’ll choose 26.



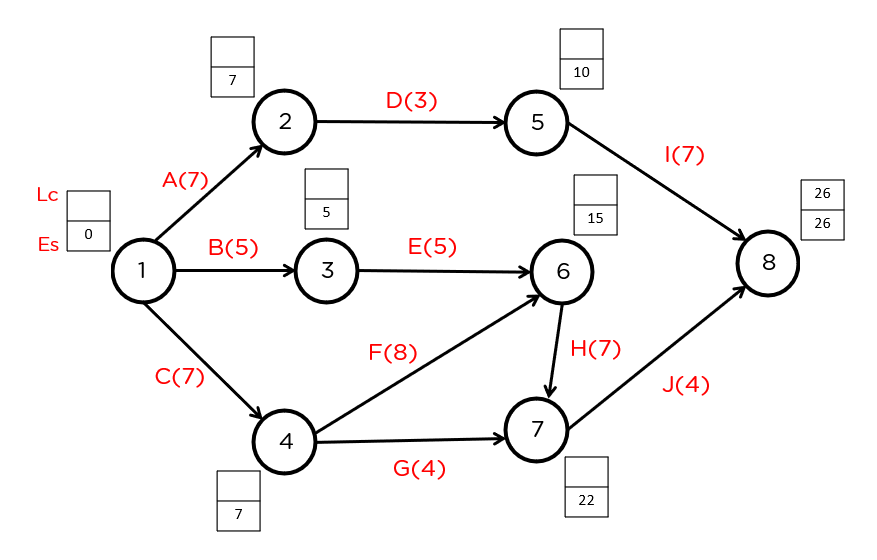
And like that, the forward pass is complete. Now, for the second part of the PERT Analysis.  Let’s take up the backward pass. For that, we will be using the following formula.

Lci = min(Lcj - Dij)

This, when put simply, means the latest completion time of the tail node is equal to the latest completion time of the head node minus the distance between the two.

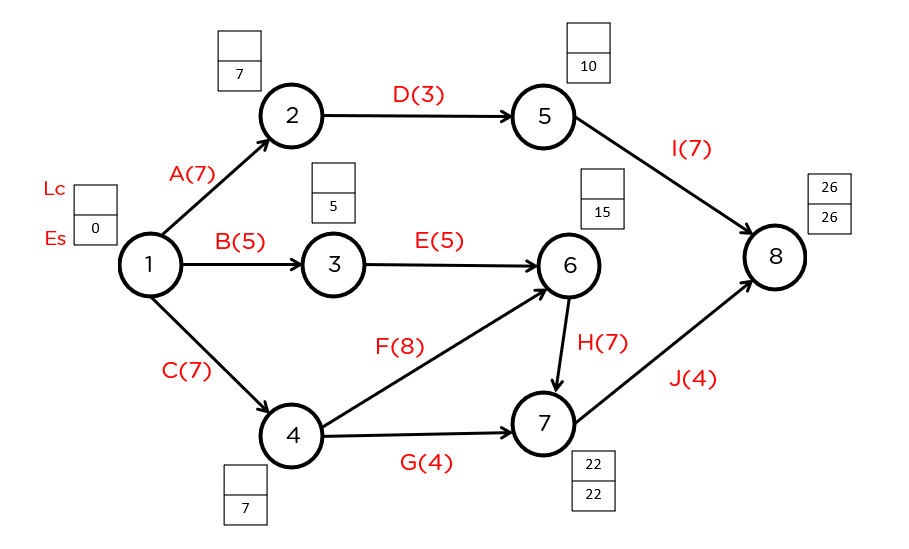
Let’s start from the final node, number 8.

The Lc for this node will always be equal to its Es.  
So, Lc8 = 26



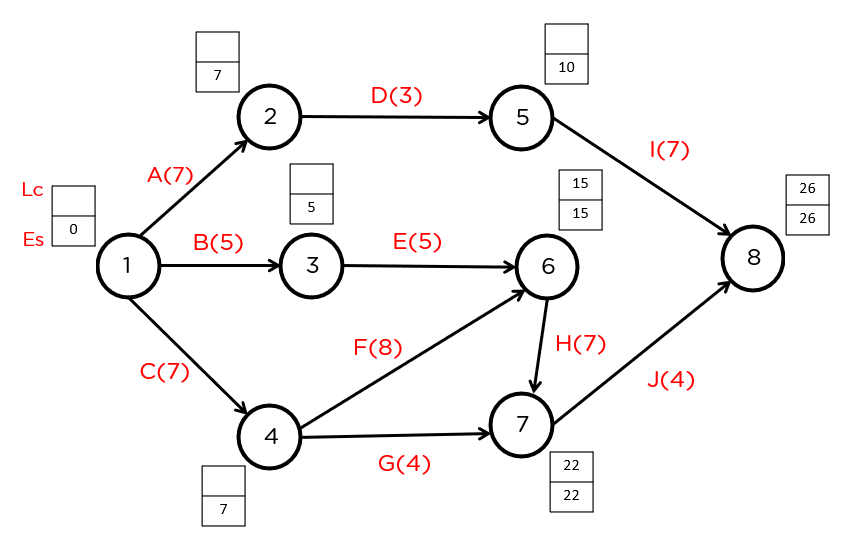
Now let’s go to node 7. Since it’s an independent node, we’ll directly apply the formula.

Lc7 = 26(Lc8) - 4(D7-8) = 22



Next up, let’s take a look at the latest completion time for node 6. Again, since it’s an independent node, we can directly apply the formula.

Lc6 = 22(Lc7) - 7(D6-7) = 15

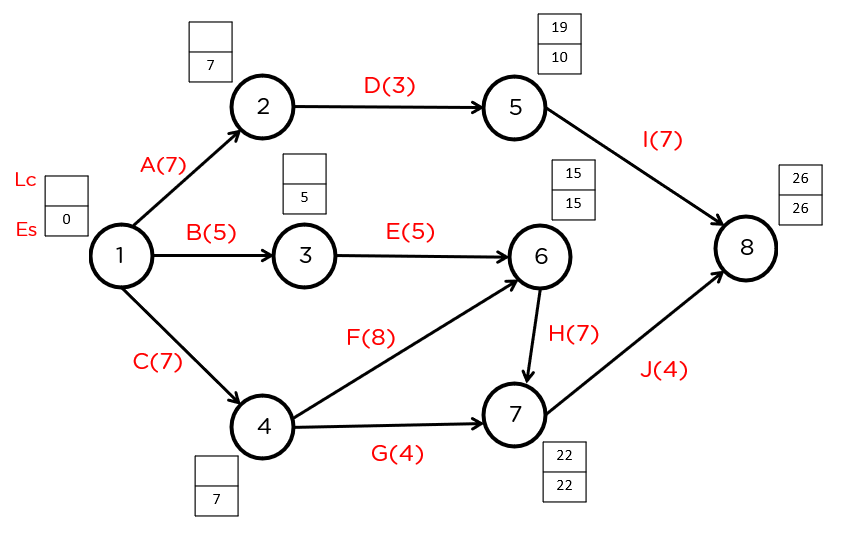


Now, for node 5.

Node 5 is an independent node. We’ll directly apply the formula here.

Lc5 = 26(Lc8) - 7(D5-8) = 19

The network diagram as part of the PERT Analysis will look like so.



Now that we’re done with node 5, let’s go to node 4.

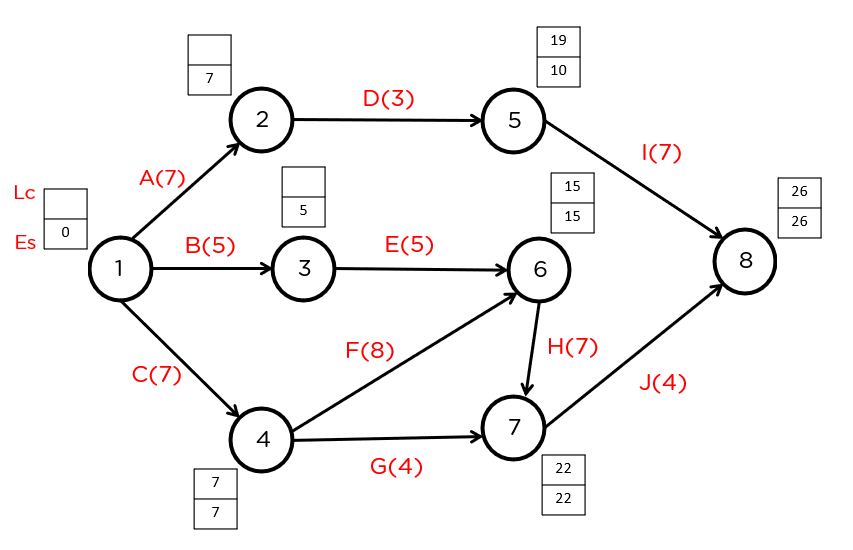
Here, we can see that two arcs connect it to nodes 6 and 7. We need to choose the minimum latest completion time from these two nodes.

Applying the formula,

Lc4 = 22(Lc7) - 4(D4-7) = 18 or

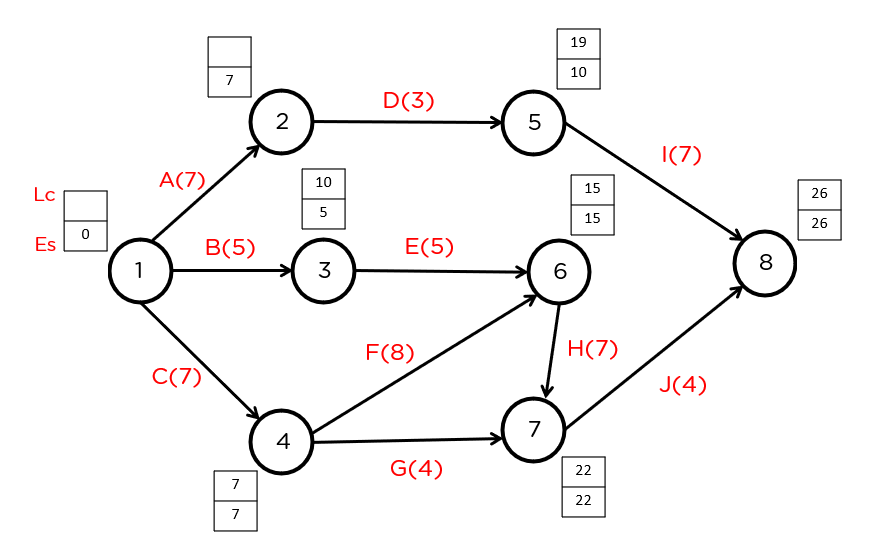
Lc4 = 15(Lc6) - 8(D4-6) = 7

Since we have to choose the minimum, we’ll choose 7.



Next, we have node 3. Since it’s an independent node with a single connection, we can directly apply the formula to it.

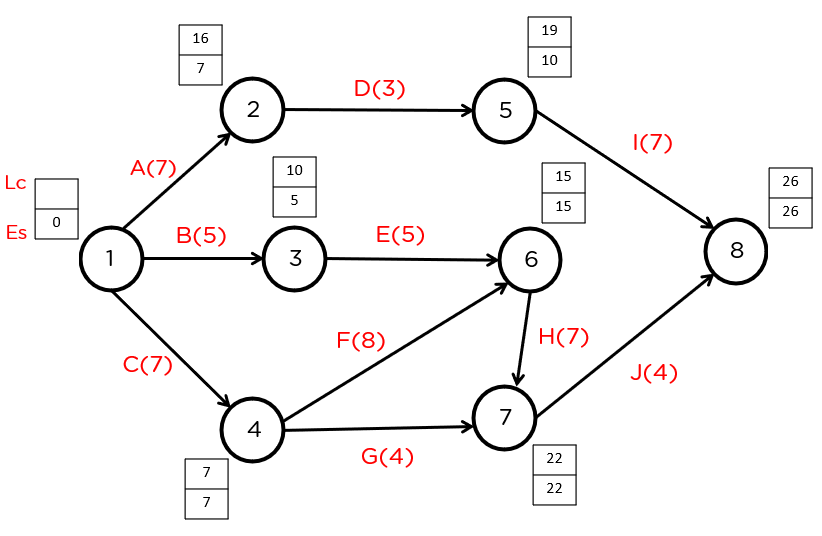
Lc3 = 15(Lc6) - 5(D3-6) = 10



Now for node 2.

We can directly apply the formula to node 2.

Lc2 = 19 (Lc5) - 3(D2-5) = 16



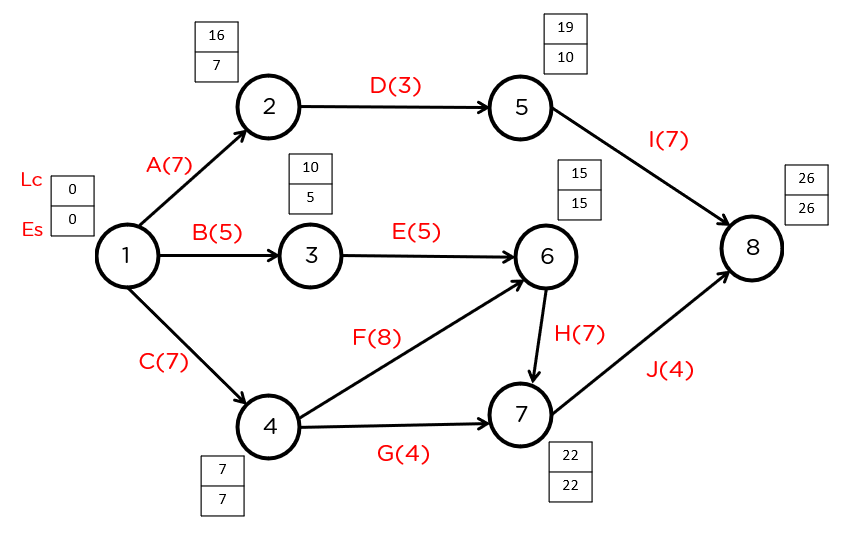
And finally, we have node 1. Since there are multiple nodes connected to node1, we’ll have to choose the minimum latest completion time.

Lc1 = 16(Lc2) - 7(D1-2) = 9 or

Lc1 = 10(Lc3) - 5(D1-3) = 5 or

Lc1 = 7(Lc4) - 7(D1-4) = 0

Since we need to choose the minimum, we’ll choose 0.



And that’s the backward pass, complete in the PERT Analysis.

Now, for the ultimate step of the critical path method. To determine the critical path, there are three major criteria that need to be satisfied.

Esi = Lci

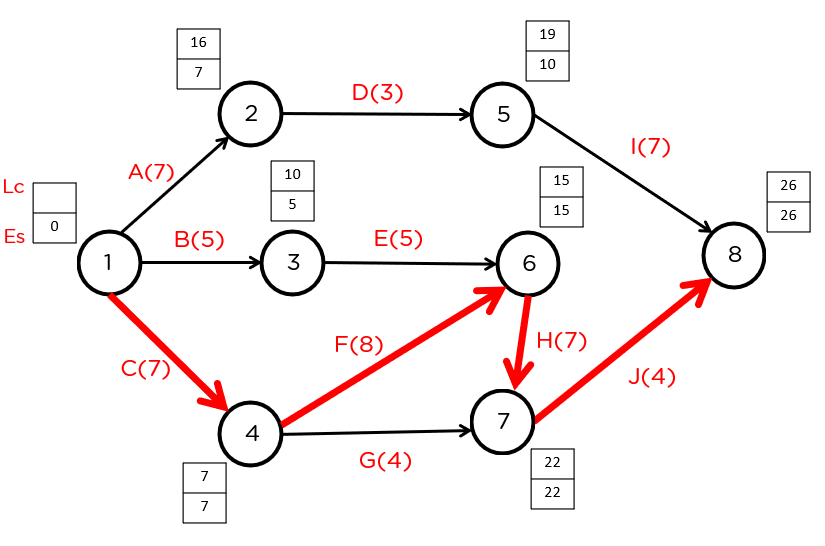
Esj = Lcj

Esj - Esi = Lcj - Lci = Dij

From the diagram, we can see that nodes that satisfy the requirements are:

1 - 4 - 6 - 7 - 8 or C - F - H - J

The estimated time is: 7 + 8 + 7 + 4 = 26 days.



**Advantages of Using a PERT Chart**

1. Visualization: PERT charts provide a visual representation of project tasks, allowing for a clear understanding of project timelines, dependencies and critical paths.
2. Planning and scheduling: PERT charts help project managers to create a detailed project plan, including defining tasks, estimating task durations, and setting deadlines.
3. Risk Management: PERT charts allow project managers to identify and analyze potential risks, enabling them to develop mitigation strategies.
4. Resource Allocation: PERT charts help project managers to identify resource requirements and allocate resources effectively, reducing the risk of delays or overloading.
5. Improved communication: PERT charts can be shared with all stakeholders, helping to ensure clear communication of project plans and progress.
6. Adaptability: PERT charts can be easily updated to reflect changes in the project, ensuring that project plans remain relevant and accurate.

**Disadvantages of Using PERT Chart**

1. Complexity: PERT charts can be challenging for those needing project management experience.
2. Time-consuming: Creating a PERT chart can be time-consuming and requires significant effort.
3. Dependent on accurate information: PERT charts rely on precise information about task durations and dependencies, and errors in this information can significantly impact the chart's effectiveness.
4. Limited scope: PERT charts are limited in their scope and may not be suitable for larger, more complex projects.
5. Over-reliance: Over-reliance on PERT charts can lead to a lack of flexibility and an inability to respond to changes in the project.
6. Inflexibility: Once a PERT chart has been created, it can be difficult to make changes to the project plan, limiting the ability to respond to changes in project requirement.

**Resource Allocation-**

To assign the available resources in an economic way is known as resource allocation. The planning of the activities and the resource required by these activities while taking into consideration both resources availability and project time is termed as resource allocation in project management.

There are 2 parts of resource allocation: Strategic Planning, and Resource Leveling. These are explained as following below.

**Strategic Planning-**

In strategic planning resource allocation is a plan for using available resources, for example human resources, specially in the near term, to achieve goals for the future. It is the process of allocating resources among various projects or business units. The strategic planning has 2 parts.

* 1. There is the basic allocation decision.
  2. There is the contingency mechanism.

The basic allocation decision is the choice of which items to fund in the plan and what level of fund in it should receive and which to leave unfunded; the resources are located to some items and not to others.There may be contingency mechanism such as priority ranking of items excluded from the plan, showing which items are to be sacrificed to reduce total funding.

**Resource Leveling-**

Resource leveling is a crucial technique in project management that aims to balance and optimize the distribution of resources over the course of a project, with the primary goal of preventing resource overloads and ensuring a smooth workflow. In the context of software project management, resource leveling involves strategically assigning available human and material resources to various tasks while considering factors such as skillsets, task dependencies, and project timelines. The process helps mitigate peaks and valleys in resource demand, ensuring that no team member is overloaded with excessive work while others remain underutilized. Resource leveling contributes to improved project efficiency, timely task completion, and overall project success. This technique often involves adjusting task schedules, priorities, or resource assignments to align with project constraints and objectives, thereby promoting a more balanced and sustainable utilization of resources throughout the project's lifecycle.

**Approach for resource allocation**

There are number of approaches to solve resource allocation problems:

1. Manual Approach
2. Algorithmic Approach
3. Combination of both
4. **Manual Approach:**

In a manual approach, project managers use their experience, expertise, and judgment to allocate resources. This often involves considering the skills and availability of team members, project priorities, and any other relevant factors.

* + *Advantages:*
    - **Flexibility:** Human judgment allows for flexibility and adaptability to unforeseen circumstances.
    - **Contextual Understanding:** Project managers can consider the broader project context and team dynamics.
  + *Considerations:*
    - **Subjectivity:** Decisions may be subjective and dependent on individual judgment.
    - **Time-Consuming:** Manual allocation can be time-consuming, especially in large and complex projects.

1. **Algorithmic Approach:**

Algorithmic approaches involve the use of mathematical models, optimization algorithms, or heuristics to automate the resource allocation process. This could include algorithms based on critical path analysis, linear programming, or other optimization techniques.

* + *Advantages:*

**Efficiency:** Algorithms can quickly analyze large datasets and provide optimized solutions.

**Objective Results:** Algoritmic approaches can offer objective, data-driven solutions.

* + *Considerations:*

**Sensitivity to Input:** The effectiveness of algorithmic approaches can be sensitive to the quality of input data and assumptions made.

**Complexity:** Some algorithms may be complex and require specialized knowledge to implement and interpret.

1. **Combination of Both:**

This approach combines manual decision-making with algorithmic support. Project managers use their judgment for certain aspects of resource allocation while relying on algorithms for others.

* + *Advantages:*

**Balanced Approach:** Combining human judgment and algorithmic analysis leverages the strengths of both approaches.

**Adaptability:** Allows for adaptability in situations where strict algorithms may not be sufficient.

* + *Considerations:*

**Integration Challenges:** Integrating manual and algorithmic approaches may require careful coordination.

**Training Requirements:** Team members may need training to understand and use any algorithmic tools effectively.

The choice of approach often depends on factors such as project complexity, team expertise, available tools, and the desired level of automation. In many cases, a hybrid approach that combines the strengths of manual decision-making with algorithmic support can offer a balanced and effective solution.

**Identifying resource requirements**

Identifying resource requirements is a crucial step in the planning phase of a software project. It involves determining the types and quantities of resources needed to successfully complete the project. Here's a systematic approach to identify resource requirements:

1. **Define Project Scope:**
   * Clearly articulate the project scope, objectives, and deliverables. Understand what needs to be accomplished to meet the project goals.
2. **Create a Work Breakdown Structure (WBS):**
   * Develop a hierarchical decomposition of the project tasks and deliverables using a Work Breakdown Structure (WBS). This helps in breaking down the project into manageable components.
3. **Identify Tasks and Activities:**
   * For each element in the WBS, identify the specific tasks and activities required to complete it. This involves a detailed analysis of what needs to be done.
4. **Determine Skill Sets:**
   * Analyze the skills and expertise required for each task. Identify the specific qualifications, knowledge, and experience needed from team members.
5. **Estimate Time and Effort:**
   * Estimate the time and effort required for each task. Consider the complexity, dependencies, and any other relevant factors that might impact resource requirements.
6. **Consider Dependencies:**
   * Identify task dependencies to understand the sequence in which tasks need to be performed. This helps in allocating resources in a logical and efficient manner.
7. **Quantify Resource Quantities:**
   * Determine the quantities of resources needed for each task. This includes personnel (developers, testers, project managers), hardware, software tools, and any other necessary resources.
8. **Assess External Dependencies:**
   * Consider external dependencies such as third-party services, contractors, or vendors. Identify the resources required from external sources and establish communication channels with them.
9. **Review Historical Data:**
   * If available, review historical project data from similar projects. This can provide insights into resource requirements and help in making more accurate estimates.
10. **Engage Stakeholders:**
    * Consult with project stakeholders, including team members, clients, and subject matter experts. Gather input and validate resource requirements to ensure a comprehensive understanding.
11. **Document Resource Requirements:**
    * Clearly document the identified resource requirements. This documentation should be detailed and serve as a reference throughout the project lifecycle.
12. **Review and Refine:**
    * Periodically review and refine resource requirements as the project progresses. Adjustments may be necessary based on changes in project scope, priorities, or unexpected challenges.

By systematically following these steps, project managers can identify resource requirements with a comprehensive understanding of the tasks involved, the skills required, and the necessary resources for successful project execution. This lays the foundation for effective resource allocation and management throughout the software development lifecycle.

Top of Form

**Resource Scheduling-**

Resource scheduling is the process of identifying when project resources are needed and allocating them based on factors such as capacity planning or resource availability. The main purpose of resource scheduling is to guarantee that there’s no over or under-allocation of resources at any point of the project.

This leads to not only getting projects done on time and within budget, but also builds morale, fosters better relationships, helps with profitability and boosts stakeholder satisfaction

In order to properly allocate resources, you need to know their availability.

**What Should You Consider Before Creating a Resource Schedule?**

As stated, the resource schedule is linked to other important aspects of your project and your organization. So, resource scheduling is a [project management](https://www.projectmanager.com/guides/project-management) decision-making activity that needs many inputs. Here are some of the things to keep in mind.

* **Resource capacity planning:** This consists of assessing the total amount of work that can be done with the resources that your organization currently has.
* **Resource utilization:** Resource utilization is a KPI that refers to the number of resources that are currently being used by your company.
* **Resource forecasting:** It’s important to estimate the future resource needs of your project. There are different [resource forecasting](https://www.projectmanager.com/blog/resource-forecasting) tools and techniques you can use to do so.
* **Resource availability:** Once you’ve understood your resource capacity planning and resource utilization, you’ll be able to determine what resources are at your disposal, or what’s your resource availability.
* **Project schedule:** Your resource schedule must be aligned with the [project schedule](https://www.projectmanager.com/guides/project-scheduling) and vice-versa. You’ll need to weigh your resource capacity, utilization and availability whenever drafting your project schedule. However, once the schedule baseline has been approved and the project execution phase begins, the resource schedule must align with the project schedule and not the other way around.

Now that we’ve learned about the main inputs that you’ll need for the resource scheduling process, let’s review some methods you can apply.

**Resource Scheduling Methods**

There are two main approaches that are followed by project managers when scheduling resources, time-constrained and resource-constrained resource scheduling. This decision is made based on resource availability.

* **Time-constrained resource scheduling:** Time constraint is a resource scheduling approach that prioritizes the timely delivery of projects even if that means extra project costs. For example, a [project manager](https://www.projectmanager.com/blog/project-manager-job-description) using this approach would hire extra workers to make up for a project schedule delay, so that deliverables can be produced on time.
* **Resource-constrained resource scheduling:** Contrary to time-constrained resource scheduling, this method builds the resource schedule based on resource availability.

**Critical path method (CPM)**

The critical path method (CPM) is a technique where you identify tasks that are necessary for project completion and determine scheduling flexibilities. A critical path in project management is the longest sequence of activities that must be finished on time in order for the entire project to be complete. Any delays in critical tasks will delay the rest of the project. CPM revolves around discovering the most important tasks in the project timeline, identifying task dependencies, and calculating task durations.

CPM was developed in the late 1950s as a method to resolve the issue of increased costs due to inefficient scheduling. Since then, CPM has become popular for planning projects and prioritizing tasks. It helps you break down complex projects into individual tasks and gain a better understanding of the project’s flexibility.

**Why use the critical path method?**

CPM can provide valuable insight on how to plan projects, allocate resources, and schedule tasks.

Here are some reasons why you should use this method:

* Improves future planning: CPM can be used to compare expectations with actual progress. The data used from current projects can inform future project plans.
* Facilitates more effective [resource management](https://asana.com/resources/resource-management-plan): CPM helps project managers prioritize tasks, giving them a better idea of how and where to deploy resources.
* Helps avoid bottlenecks: Bottlenecks in projects can result in lost valuable time. Plotting out project dependencies using a network diagram, will give you a better idea of which activities can and can’t run in parallel, allowing you to schedule accordingly.

**How to find the critical path?**

Finding the critical path involves determining the longest sequence of dependent and linked tasks in a project that must be completed on time to ensure the project stays on schedule. The critical path has zero slack or float, meaning any delay in tasks along this path will directly impact the project's completion date. Here's a step-by-step guide on how to find the critical path using the Critical Path Method (CPM):

**Step 1: Create a Project Network Diagram**

1. List all the tasks in the project.
2. Identify task dependencies. Determine which tasks must be completed before others can start.

**Step 2: Define Task Durations**

Assign a duration to each task, indicating the time required to complete it. This can be in days, weeks, or any unit relevant to your project.

**Step 3: Calculate Earliest Start (ES) and Earliest Finish (EF)**

For each task, calculate the Earliest Start (ES) and Earliest Finish (EF) based on task durations and dependencies. Use the following formulas:

* **ES (Earliest Start):** The earliest time a task can start based on its dependencies.
* **EF (Earliest Finish):** The earliest time a task can finish based on its ES and duration.

**Step 4: Calculate Latest Start (LS) and Latest Finish (LF)**

Working backward, calculate the Latest Start (LS) and Latest Finish (LF) for each task. Use the following formulas:

* **LS (Latest Start):** The latest time a task can start without delaying the project.
* **LF (Latest Finish):** The latest time a task can finish without delaying the project.

**Step 5: Identify Slack or Float**

Slack or float is the total time that a task can be delayed without delaying the project. Tasks on the critical path have zero slack. Identify tasks with zero slack to determine the critical path.

**Step 6: Determine the Critical Path**

The critical path is the longest path through the project network where the total duration is equal to the project duration. It is composed of tasks with zero slack. Trace the path with zero slack from the project start to finish.

**Example Project:**

Consider the following tasks for a software development project:

1. **Task A: Requirements Gathering (5 days)**
2. **Task B: System Design (10 days)**
3. **Task C: Development (15 days)**
4. **Task D: Testing (8 days)**
5. **Task E: Deployment (5 days)**
6. **Task F: User Training (7 days)**

Now, let's establish task dependencies:

* Task B (System Design) can only start after Task A (Requirements Gathering) is completed.
* Task C (Development) can only start after Task B (System Design) is completed.
* Task D (Testing) can only start after Task C (Development) is completed.
* Task E (Deployment) can only start after Task D (Testing) is completed.
* Task F (User Training) can only start after Task E (Deployment) is completed.

**Step 1: Create a Network Diagram**

Construct a network diagram representing tasks, durations, and dependencies. Here's the diagram:

A(5) \ B(10) \ C(15) \ D(8) \ E(5) \ F(7)

**Step 2: Calculate ES and EF**

Calculate the earliest start (ES) and earliest finish (EF) for each task:

* For Task A: ES = 0, EF = 5
* For Task B: ES = 5, EF = 15
* For Task C: ES = 15, EF = 30
* For Task D: ES = 30, EF = 38
* For Task E: ES = 38, EF = 43
* For Task F: ES = 43, EF = 50

**Step 3: Calculate LS and LF**

Calculate the latest start (LS) and latest finish (LF) for each task by working backward:

* For Task F: LS = 43, LF = 50
* For Task E: LS = 38, LF = 43
* For Task D: LS = 30, LF = 38
* For Task C: LS = 15, LF = 30
* For Task B: LS = 5, LF = 15
* For Task A: LS = 0, LF = 5

**Step 4: Identify Slack (Total Float)**

Calculate slack for each task using the formula: Slack = LF - EF or LS - ES. Tasks with zero slack are on the critical path.

* For Task A: Slack = 5 - 5 = 0
* For Task B: Slack = 15 - 15 = 0
* For Task C: Slack = 30 - 30 = 0
* For Task D: Slack = 38 - 38 = 0
* For Task E: Slack = 43 - 43 = 0
* For Task F: Slack = 50 - 50 = 0

**Step 5: Identify the Critical Path**

The critical path is the sequence of tasks with zero slack. In this example, the critical path is A -> B -> C -> D -> E -> F.

**Difference between PERT and CPM :**

|  |  |
| --- | --- |
| **PERT** | **CPM** |
| PERT is that technique of project management which is used to manage uncertain (i.e., time is not known) activities of any project. | CPM is that technique of project management which is used to manage only certain (i.e., time is known) activities of any project. |
| It is event oriented technique which means that network is constructed on the basis of event. | It is activity oriented technique which means that network is constructed on the basis of activities. |
| It is a probability model. | It is a deterministic model. |
| It majorly focuses on time as meeting time target or estimation of percent completion is more important. | It majorly focuses on Time-cost trade off as minimizing cost is more important. |
| It has Non-repetitive nature of job. | It is appropriate for reasonable time estimation. |
| It is appropriate for high precision time estimation. | It is appropriate for reasonable time estimation. |
| There is no chance of crashing as there is no certainty of time. | There may be crashing because of certain time boundation. |
| It doesn’t use any dummy activities. | It uses dummy activities for representing sequence of activities. |
| It is suitable for projects which required research and development. | It is suitable for construction projects. |

**Gantt Chart-**

**Generalized Activity Normalization Time Table (GANTT) chart** is type of chart in which series of horizontal lines are present that show the amount of work done or production completed in given period of time in relation to amount planned for those projects. It is horizontal bar chart developed by Henry L. Gantt (American engineer and social scientist) in 1917 as production control tool. It is simply used for graphical representation of schedule that helps to plan in an efficient way, coordinate, and track some particular tasks in project.

The purpose of Gantt chart is to emphasize scope of individual tasks. Hence set of tasks is given as input to Gantt chart. Gantt chart is also known as timeline chart. It can be developed for entire project or it can be developed for individual functions. In most of projects, after generation of timeline chart, project tables are prepared. In project tables, all tasks are listed in proper manner along with start date and end date and information related to it.

**Gantt chart represents following things :**

* All the tasks are listed at leftmost column.
* The horizontal bars indicate or represent required time by corresponding particular task.
* When occurring of multiple horizontal bars takes place at same time on calendar, then that means concurrency can be applied for performing particular tasks.
* The diamonds indicate milestones.

**Advantages :**

* **Simplify Project –**   
  Gantt charts are generally used for simplifying complex projects.
* **Establish Schedule –**   
  It simply establishes initial project schedule in which it mentions who is going to do what, when, and how much time it will take to complete it.
* **Provide Efficiency –**   
  It brings efficiency in planning and allows team to better coordinate project activities.
* **Emphasize on scope –**   
  It helps in emphasizing i.e., gives importance to scope of individual tasks.
* **Ease at understanding –**   
  It makes it easy for stakeholders to understand timeline and brings clarity of dates.
* **Visualize project –**   
  It helps in clearly visualizing project management, project tasks involved.
* **Organize thoughts and Highly visible –**   
  It organizes your thoughts and can be highly visible so that everyone in enterprises can have basic level of understanding and have knowledge about what’s happening in project even if they are not involved in working.
* **Make Practical and Realistic planning –**   
  It makes the project planning practical and realistic as realistic planning generally helps to avoid any kind of delays and losses of many that can arise.

**Disadvantages :**

* Sometimes, using Gantt chart makes project more complex.
* The size of bar chart dost not necessarily indicate amount of work done in project.
* Gantt charts and projects are needed to be updated on regular basis.
* It is not possible or difficult to view this chart on one sheet of paper. The software products that produce Gantt chart needed to be viewed on computer screen so that whole project can be seen easily.

**Applications :**

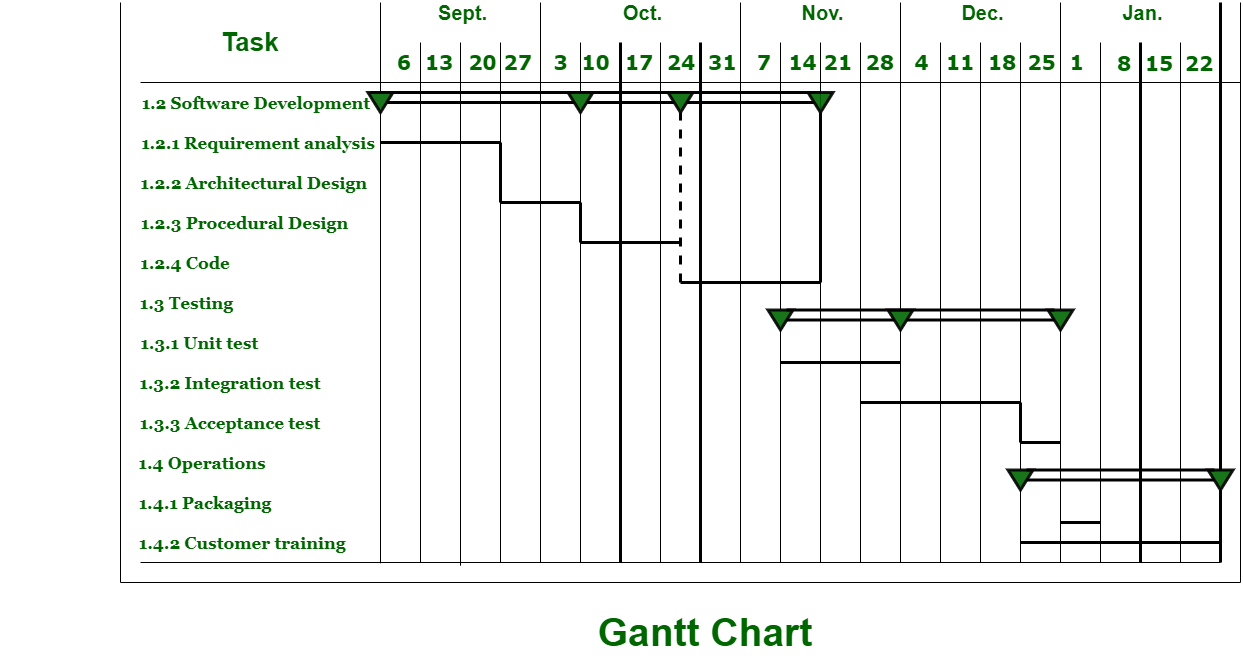
There are several professions, where use of gantt chart is very beneficial. Some of them are given below:

**Advertising Manager –** Advertising Managers generally controls and supervises end result of advertising companies, scheduling advertisements in different media, etc.

**Operations Manager –** Operations Managers generally control and handle resources that are essential for company operations.

**Project Manager –** Project Managers generally motivates project teams, collaborate with team members, schedule task and complete that on time, and report to stakeholders, etc.

**Example :**   
Nowadays, there are many companies and teams that use Gantt chart to plan, schedule, and execute their projects. Some of them are consulting agencies, manufacturing companies, Marketing teams, Construction companies, etc. Below is an example of Gantt chart:



**Staffing -**

Staffing is the art of acquiring, developing, and maintaining a satisfactory and satisfied workforce. Staffing is that function by which a manager builds an organization through the recruitment, selection, and development of the individual, which also includes a series of activities. It ensures that the organization has the right number of people at the right places, at the right time, and performing the right thing.

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. Therefore, these are called elements or steps of the staffing process.

* **Manpower Planning:** 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.The difference between workload and workforce is calculated to determine shortage and surplus of manpower. Excess workload indicates understaffing, i.e., the need of appointing more people and excess workforce indicates overstaffing, i.e., need to remove or transfer some employees to other places.
* **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.
* **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. There are many steps involved in the process of employee selection. These steps include preliminary screening, filling-in application, written test, interviews, medical examination, checking references, and issuing a letter of appointment to the candidates. The most suitable candidates who meet the requirement of the vacant job are selected. 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.
* **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. Correct placement helps to reduce labour turnover and absenteeism. Here, orientation means introducing new employees to the organization. It is the process of introducing and familiarizing newly appointed candidates with their job, work groups and the organization so that they may feel at home in the new environment.
* **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. By providing such opportunities to its employees for career advancement, the organization captivates the interest and holds on of its talented employees. The majority of the organization has a distinct department for this purpose, that is, the Human Resource Department. Though in small organizations, the line manager has to do all the managerial functions viz, planning, organizing, staffing, controlling, and directing. The process of staffing further involves three more stages.
* **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. The employee should be well aware of his standards and his superior is responsible for proving feedback on his performance. The process of performance appraisal, thus includes specifying the job, performing appraisal performance, and providing feedback.
* **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. Promotions are an essential part of any employee’s career. Promotion refers to the transferring of employees from their current positions to a higher level increasing their responsibilities, authority and pay.
* **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. The worth of the job needs to be decided. Therefore, 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.