

* Software Project Management

Software Project Management encompasses a range of activities and methodologies aimed at planning, organizing, executing and controlling software development projects. It involves coordinating resources, timelines, budgets, and tasks to ensure that projects are completed successfully and deliver value to stakeholders.

* Project and Software Project

A project is a temporary endeavor undertaken to create a unique product, service, or result. It has a defined beginning and end, and typically involves a set of tasks, resources, and constraints aimed at achieving specific objectives.

A software project is a specific type of project that focuses on the development of software products or solutions. This includes activities such as software design, coding, testing and deployment.

* Similarities between Project & Software Project

1) Temporary Nature

Both projects and software projects are temporary endeavors with defined beginnings and ends.

2) Goal Oriented

Both types of projects are goal-oriented, aiming to deliver a unique product or service.

3) Resource Allocation

Projects, including software projects require the allocation of resources such as time, money, manpower, and materials.

4) Risk Management

Both types of project involve identifying, assessing and managing risks to mitigate potential negative impacts on project success.

* Differences between Project and Software Project

1) Product vs Service

While projects can involve the creation of a wide range of deliverables, software project specifically focus on the development of software products or solutions.

2) Iterative Development

Unlike some traditional projects that follow a linear progression from initiation to closure, software projects often adopt iterative or incremental development.

3) Change Management

Software projects frequently encounter changes in requirements, technology or stakeholder priorities usually more than in traditional projects.

1) Productivity

Software project productivity varies greatly with change in technology or workers whereas productivity doesn't vary much in traditional project.

2) Leadership

Software Project needs leaders and managers, not just administrators whereas a capable administrator is enough to run an ordinary project.

* Importance of Software Project Management

1) Achieving Objectives

Software Project Management ensures that project meet their goals and deliverables as defined by stakeholders.

2) Managing Resources

It involves effectively allocating and tracking resources like time, budget, and manpower to ensure project success.

3) Mitigating Risks

Project managers identify and manage potential risks, reducing the chance of project failures or delays.

4) Ensuring Quality

Quality management processes guarantee that software products meet desired standards and are free of defects.

5) Facilitating Communication

Effective communication among team members and stakeholders ensures alignment and engagement throughout the project.

6) Managing Change

Project managers handle change in requirements, technology or priorities keeping projects on track while adapting to evolving needs.

7) Adopting Best Practices

Leveraging project management methodologies and tools improves efficiency and consistency across projects.

8) Meeting Deadlines

Software Project Management helps set realistic schedules, monitor progress, and address issues to ensure projects are completed on time.

* Problems in Software Project Management

1) Unclear Requirements

Incomplete or ambiguous requirements can lead to misunderstandings, scope creep and delays in project delivery.

2) Poor Planning

Inadequate project planning, including inaccurate estimations, unrealistic deadlines, and insufficient resource allocation, can result in project delays, budget overruns, and quality issues.

3) Scope Creep

Uncontrolled changes to project scope, often driven by evolving requirements or stakeholder requests can disrupt project timelines, increase costs, and strain resources.

4) Resource Constraints

- Limited availability of skilled personnel, budgetary constraints, or technological limitations can impact project execution and quality, requiring careful resource management and allocation.

5) Technical Challenges

- Complex technical requirements, compatibility issues or unexpected technical hurdles can result in delays, rework and compromised project quality.

6) Risk Management

- Inadequate identification, assessment and mitigation of project risks can lead to project failures, cost overruns, or missed deadlines.

* Processes of Software Project Management

1) Initiation

- Define project objectives, scope & deliverables
- Identify stakeholders and establish project governance.
- Conduct feasibility studies and assess project risks
- Develop a project charter or initiation document.

2) Planning

- Develop a detailed project plan outlining tasks, timelines, milestones and resource requirements
- Define a project scope, requirements and acceptance criteria
- Estimate costs, budget and allocate resources
- Develop risk management and quality management plans
- Define role and responsibilities within the project team.

3) Execution

- Implement the project plan by executing tasks according to the defined schedule
- Manage project resources, including personnel, budget and materials
- Monitor progress against the project plan and track key performance indicators (KPIs)
- Address issues, risks and changes as they arise

4) Monitoring and Controlling

- Continuously monitor project performance, quality and progress.
- Compare actual results to planned objectives and identify variances
- Manage changes to project scope, requirements, or schedules through formal change control processes
- Ensure compliance with relevant standards, regulations and organizational policies.

5) Closure

- Complete all project deliverables and obtain stakeholder acceptance.
- Conduct a final project review to assess outcomes, lessons learned, and areas for improvement.
- Close out contracts, procurements and ~~final~~ financial accounts related to the project.
- Conduct post-implementation reviews to evaluate the project's long-term impact and benefits.

* Characteristics of a Good Project Manager

1) Effective Communication

Able to convey information clearly and listen actively, fostering open dialogue among team members and stakeholders.

2) Strong Leader

Provides direction, motivation and guidance to the team, making decisions decisively and handling conflicts professionally.

3) Highly Organized

Manages tasks, resources and timelines efficiently, ensuring that projects stay on track and objectives are met.

4) Skilled Problem Solver

Identifies issues, analyzes root causes, and develops practical solutions to overcome challenges and obstacles.

5) Adaptable to Change

Comfortable navigating uncertainty and adjusting plans and strategies in response to evolving circumstances

6) Team builder

Creates a collaborative and inclusive team culture, promoting trust, respect and teamwork among members.

7) Expert in Stakeholder management

Engages and communicates with stakeholders throughout the project lifecycle, managing their interests and expectations.

* Software Process Scheduling

* Objectives of activity planning

1) Feasibility Assessment

Is the project possible with required timescales and resource constraints?

2) Resource Allocation

What are the most effective ways of allocating resources to the project. When should the resources be available.

3) Detailed costing

How much will the project cost and when is that expenditure likely to take place?

4) Motivation

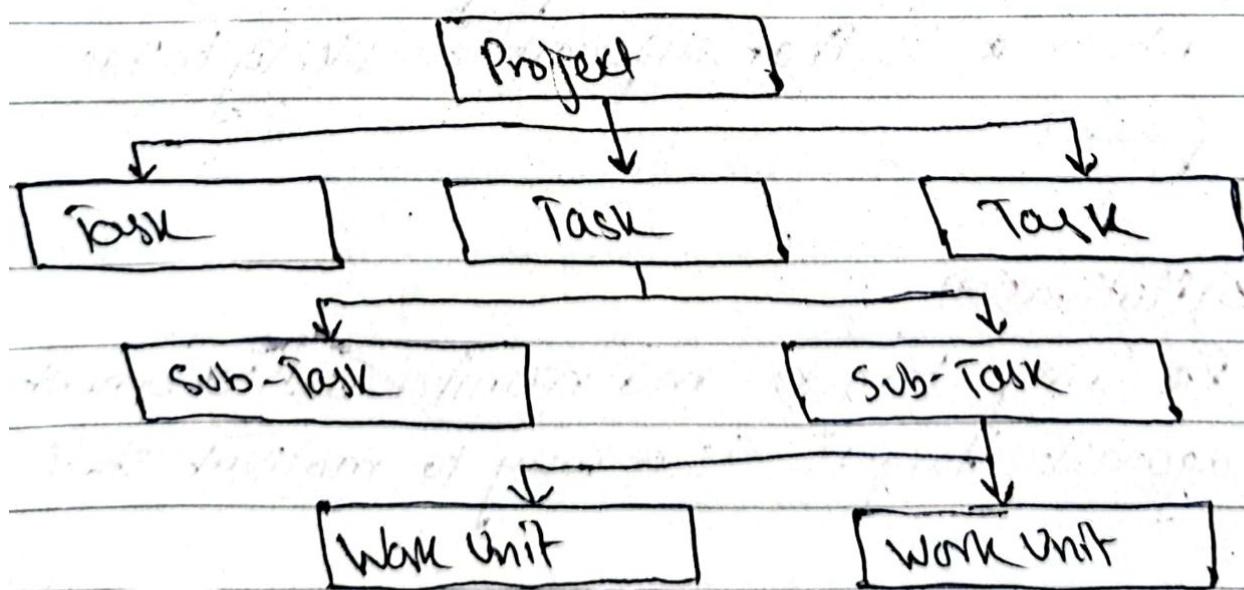
Providing targets and monitor achievement against targets is a way to motivate staff

5) Coordination

Staffs in diff departments should work together.

* Work Breakdown Structure (WBS)

A Work Breakdown Structure (WBS) includes dividing a large and complex project into simpler, manageable, and independent tasks. The root of this tree structure is labeled by the project name itself. For constructing a work breakdown structure, each node is recursively decomposed into smaller sub-activities, until at the leaf level, the activities become undividable and independent. It follows a top-down approach.



* Characteristics of WBS

- Hierarchical
- Deliverable Oriented
- Decomposition
- Accurate Time and Cost Estimation
- Visual Representation

* Pros of WBS

- Clarity and Organization
- Scope Management
- Resource allocation
- Risk Management

* Cons of WBS

- Time consuming
- Overemphasises on structure
- Scope changes
- Limited flexibility
- Maintenance

Network Programming Model:

* Critical Path Model (CPM)

A critical path is a project management technique used to determine the longest sequence of dependent tasks and calculate the shortest possible duration for completing the project.

Critical path method uses a network diagram to represent the project tasks and their dependencies. Tasks are represented as nodes, and the dependencies between them are represented as directed arrows.

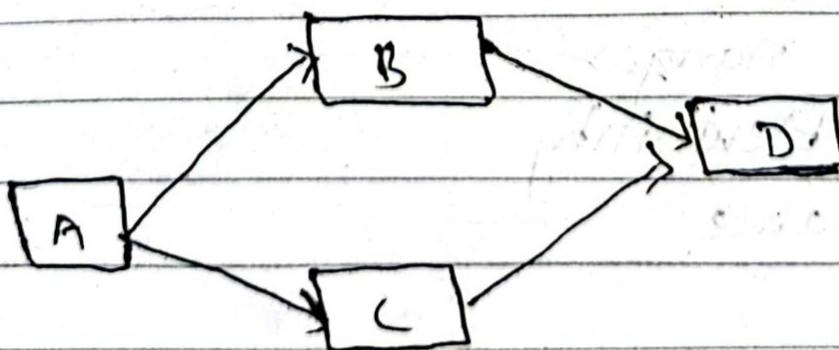


Fig: Critical Path Method

* Characteristics of CPM

1) Task Dependency Analysis

CPM involves analyzing task dependencies—the sequence in which activities must be completed.

2) Network Diagram Representation

CPM typically uses a network diagram to visualize project tasks and their relationships.

3) Duration estimation

By analyzing the critical path and task dependencies, CPM provides an estimation of the project's duration.

4) Risk Management

Through the identification of critical path analysis, CPM assists in risk management by highlighting areas where delays could impact the project timeline.

* Pros of CPM

- Clear Visualization
- Efficient Resource Allocation
- Accurate Duration Estimation
- Effective Schedule Management
- Optimization of Project Resources

* Cons of CPM

- Complexity
- Sensitivity to Changes
- Limited Flexibility
- Resource Constraints
- Time Consuming
- Software Dependency

* Program Evaluation and review technique (PERT)

PERT is a program management technique that uses probabilistic time estimates for tasks, considering optimistic, pessimistic, and most likely scenarios. It helps estimate project duration, identify critical paths, and manage project risks effectively.

PERT requires three estimates of time for each task:

- 1) Optimistic (O)
- 2) Pessimistic (P)
- 3) Most likely (M)

These estimates are used to calculate the expected duration (TE) for each task using the formula:

$$TE = \frac{O + 4M + P}{6}$$

* Characteristics of PERT

- Probabilistic Time Estimates
- Expected Duration Calculation
- Critical Path Analysis
- Flexibility
- Resource Allocation Consideration

* Pros of PERT

- Resource Optimization
- Visualization
- Flexibility
- Risk Management

* Cons of PERT

- Complexity
- Resource Allocation Limitations
- Overemphasis on Critical Paths
- Risk of Inaccurate Estimates

* Software Effort Estimation

Effort estimation is a crucial aspect of software project management as it helps in planning, scheduling and resource allocation. Various techniques for software effort estimation

1) Expert Judgement

This involves seeking input from experienced individuals or teams who have worked on similar projects in the past. They provide estimates based on their knowledge and experience.

2) Analogies Estimation

Also known as top-down estimation, this technique involves using historical data from similar past events to estimate effort required.

3) Parametric Estimation

This method uses statistical relationship between historical data and project parameters.

4) Algorithmic Models

These models use mathematical algorithms to estimate effort based on various project parameters and factors.

5) Expert Estimation

This technique involves breaking down the project into smaller tasks and having experts estimate the effort required for each task.

6) 3-Point Estimation

This technique involves estimating three scenarios for each task; optimistic, pessimistic, and most likely. The average of these estimates is then used to calculate the overall effort.

7) Estimation by analogy

* Problems of over estimation:

- 1) Wastage of Resources
- 2) Reduced Competitiveness
- 3) Unrealistic Deadlines
- 4) Delayed Time-to-Market
- 5) Loss of Trust

* Problems of under estimation:

- 1) Project Delays
- 2) Budget Overruns
- 3) Quality Compromises
- 4) Employee Burnout
- 5) Reputation Damage

* Estimating by Analogy (Top-Down approach)

Analogous technique is a technique where you predict the time required for a project based on the similar project in the past.

For instance, if you saw that it takes your team six months on average to create a mobile wallet app, you'd give roughly same estimation for any mobile app in the future.

Analogous estimation is the best route if you don't have enough time and data available to give a more accurate timeframe. It's also relatively easy, requiring no fancy calculations.

However its biggest flaw is accuracy. That's because analogous estimation is based on the assumption that the project exactly the same as your current project.

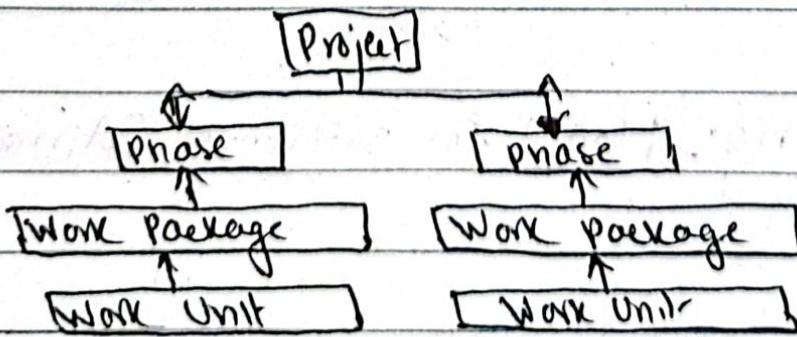
* Bottom-up Estimating

Dividing the whole project into activities, then calculate cost/schedule of every activity, aggregate cost/schedule of every activity to the project level.

For instance, let's say you determine the project requires roughly 50 individual tasks, each requiring an average of four days to complete.

Thus, your software should take 200 days to complete, or around 6-7 months.

Work Breakdown Structure (WBS) is an example of this approach.



* Parametric models

Parametric estimation is the method of estimating the effort, time or cost required for a software project based on statistical relationships between project parameters and historical units. Parametric estimation uses mathematical models to predict parameters based on specific project requirements.

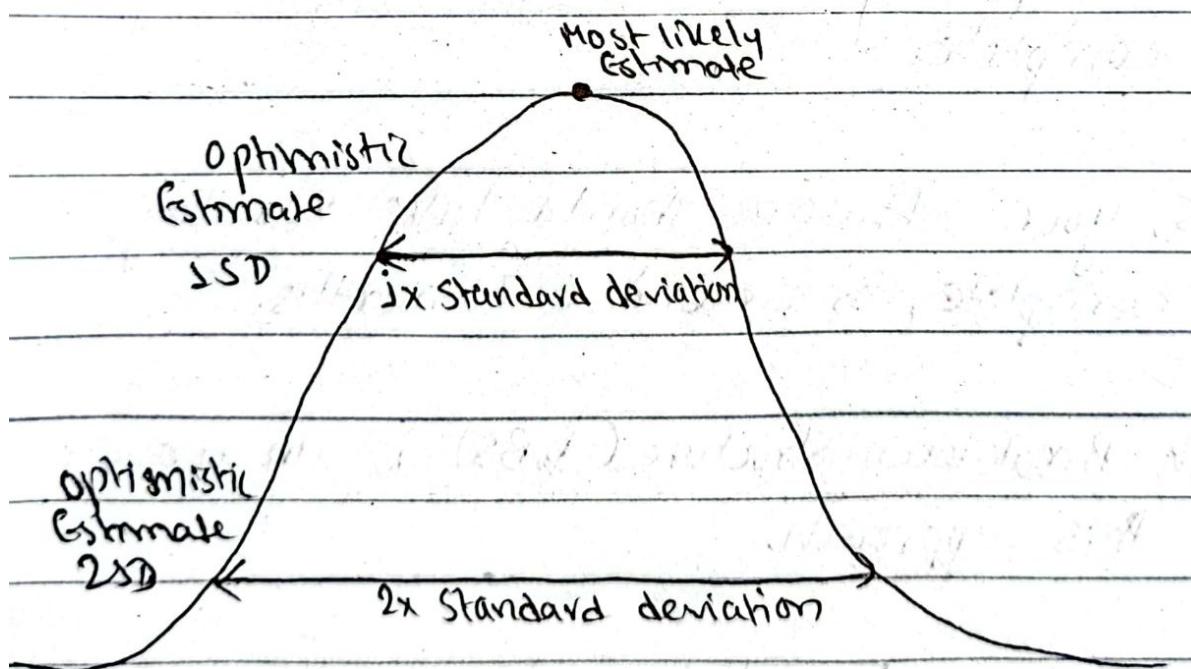


fig. Model for software Estimation

* Advantages of Parametric Models

- Data - Driven Approach
- Accuracy
- Efficiency
- Standardization
- Scalability
- Early Estimation

* Dis - advantages of Parametric Models

- Data Requirements
- Complexity
- Model Maintenance
- Risk of Biases
- Limited Flexibility

* Cost-Benefit Analysis

Cost-Benefit Analysis (CBA) is a systematic approach used in software project management to evaluate the potential costs and benefits associated with a project. It helps project stakeholders make informed decisions by comparing the expected costs of implementing a project with its expected benefits.

* Pros

- Objective Decision Making
- Stakeholder Alignment
- Risk Management
- Resource Allocation

* Cons

- Qualification changes
- Subjectivity
- Time and effort
- Risk of overlooking Intangible factors

* Cash flow forecasting

Cash flow forecasting is a vital aspect of financial management in software project management, enabling organizations to anticipate and manage their cash inflows and outflows over a specific period.

* Steps involved in cash flow forecasting

- 1) Identify cash inflows
- 2) Estimate cash outflows
- 3) Forecast timing
- 4) Build a cash flow projection
- 5) Review and update regularly.

* Cost Benefit evaluation techniques

1) Net-Present Value (NPV)

- NPV calculates the present value of all cash inflows and outflows associated with a project, discounted at a specified rate.

$$NPV = \sum \frac{\text{Cashflow}}{(1+r)^t}$$

2) Return on Investment (ROI)

ROI measures the ratio of net benefits to the initial investment in the project.

$$ROI = \frac{\text{Net Profit}}{\text{Investment}} \times 100$$

* Total Quality Management (TQM)

TQM is the integration of all functions and processes within an organization in order to achieve continuous improvement of the quality of goods and services.

* Benefit of TQM

- 1) Greater customer loyalty
- 2) Market share improvement
- 3) Higher stock prices
- 4) Reduced service calls
- 5) Higher prices
- 6) Improve quality
- 7) Employment participation.

* Six Sigma

It is a set of techniques and tools for process improvement.

It seeks to improve the quality of process outputs by identifying and removing the causes of defects.

1) **Define**: Clearly outline project goals and customer requirements.

2) **Measure**: Gather data to assess project performance and identify areas for improvement.

3) **Analyze**: Use statistical methods to understand root causes of issues.

4) **Improve**: Implement solutions to address identified problems.

5) Control: Establish processes to maintain improvements.

6) Verify: Verify the effectiveness of implemented changes and ensure they meet project objectives.

* Software Quality & Importance

Software quality refers to the degree to which a software product meets the specified requirements and user expectations.

* Key factors involving software quality

1) Requirements measurement

2) Software design

3) Code quality

4) Testing and quality assurance

5) Defect management

* Importance of Quality Software

- 1) Software Satisfaction
- 2) Business Reputation
- 3) Cost Savings
- 4) Risk Mitigation
- 5) User Productivity

* ISO9126

ISO9126 is an international standard that defines a framework for evaluating the quality of software products. It provides a comprehensive set of quality characteristics and sub-characteristics that can be used to assess various aspects of software quality.

* Characteristics of ISO9126 model

1) Functionality

- Suitability
- Accuracy
- Interoperability
- Compliance
- Security

2) Reliability

- Maturity
- Fault tolerance
- Recoverability
- Reliability

3) Usability

- Understandability
- Learnability
- Operability
- Attractiveness
- Compliance

4) Efficiency

- Time behavior
- Resource utilization

5) Maintainability

- Analyzability
- Changeability
- Stability
- Testability

6) Portability

- Adaptability
- Installability
- Co-existence
- Replaceability

* Software Configuration Management (SCM)

Software Configuration Management (SCM) is the process of managing and controlling changes to software artifacts throughout the software development lifecycle. It involves establishing and maintaining consistency, integrity and traceability of software components, configurations and documentation.

* SCM Processes are 8-

- 1) Identification
- 2) Version Control
- 3) Change Control
- 4) Configuration Audit
- 5) Reporting

* Calculate the Net present value (NPV) of the company.

Cost of Investment = Rs 2,00,000

Total time = 5 years

Rate of discount = 10%

Year	Cash flows (CF)	IF ($R=10\%$)	PV
0	-200000	1	-200000
1	50000	0.909	45450
2	50000	0.826	41300
3	60000	0.751	45060
4	60000	0.683	40980
5	60000	0.621	37260

$$IF = \frac{1}{(1+r)^t}, PV = IF \times CF$$

$$\text{So, } NPV = \sum PV = \text{Rs } 10050$$

* Calculate ROI for the following project.

Year	Project
0	-50000
1	5000
2	5000
3	20000
4	40000
5	5000

Solutions

$$\text{Total Profit} = 70000$$

Investment Period = 5 years

$$\text{Total Investment} = 50000$$

Now, $\text{ROI} = \frac{\text{average profit}}{\text{Total Investment}} \times 100\%$

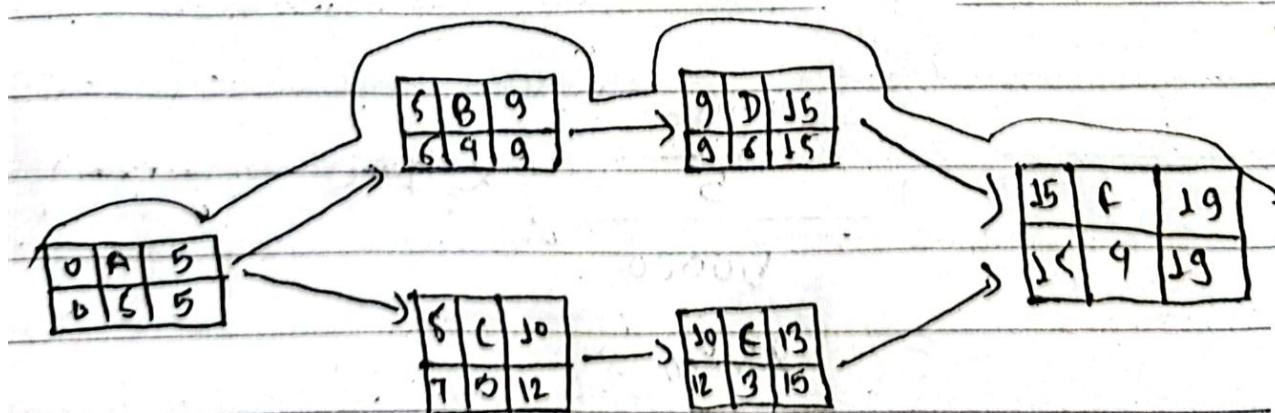
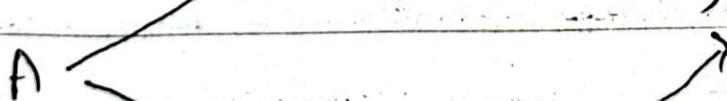
$$= \frac{70000}{5} \times 100$$

~~Total Profit~~

$$= 28\%$$

* Using CPM, identify the critical path of a project.

Activity	Predecessor	Duration
A	-	5
B	A	4
C	A	5
D	B	6
E	C	3
F	D, E	4

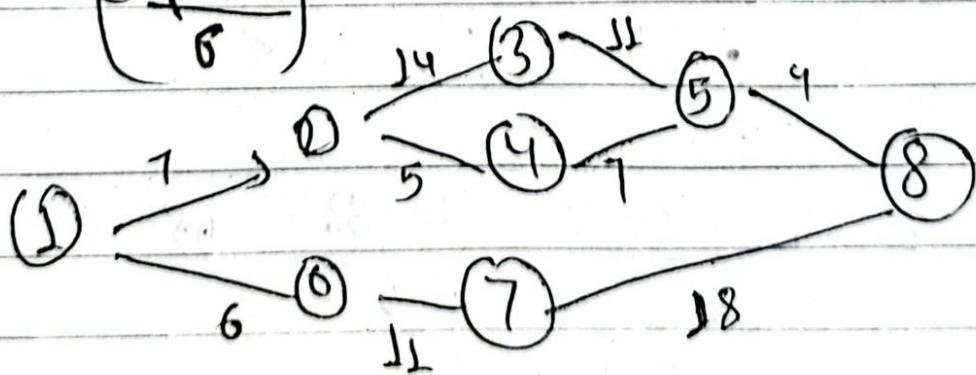


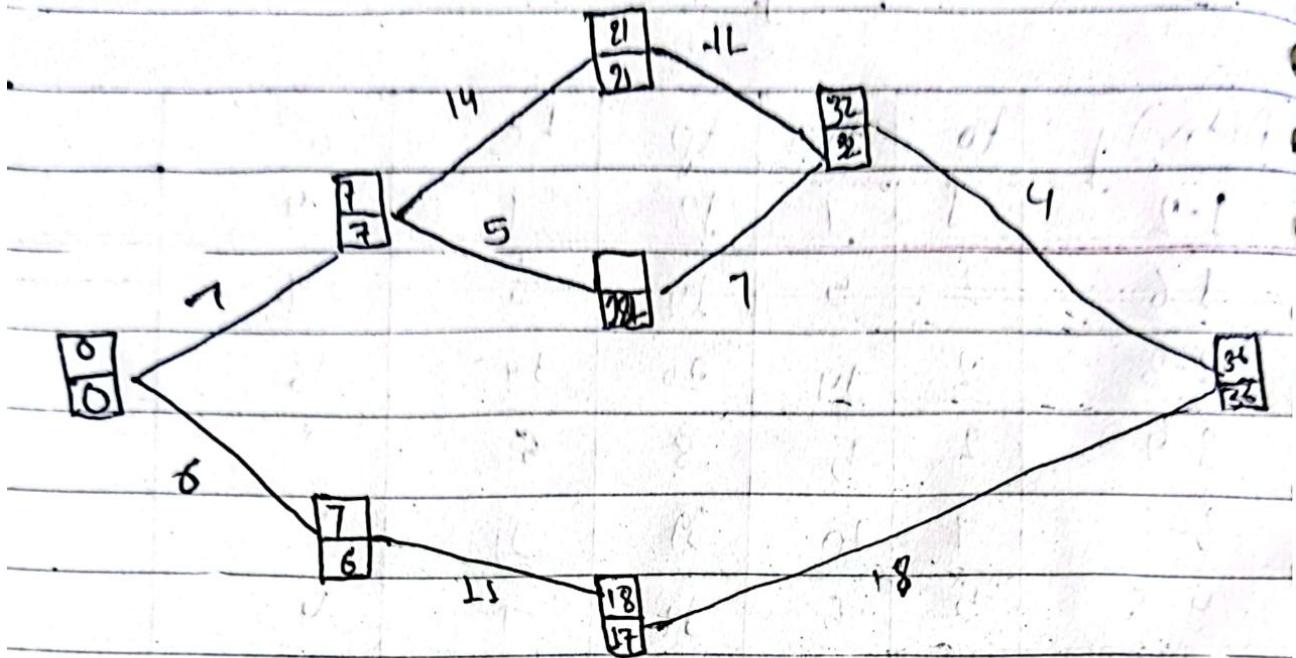
* Calculate using PERT

Activity	to	tm	tp	te	σ^2	
1-2	1	7	13	7	4	
1-6	2	5	14	6	4	
2-3	2	14	26	14	16	
2-4	2	5	8	5	1	
3-5	7	10	19	21	4	
4-5	5	5	17	7	4	
6-7	5	8	29	21	16	
5-8	3	3	9	4	1	
7-8	8	27	32	18	16	

$$Te = \frac{0_e + P_e + M_e}{6}$$

$$\sigma^2 = \left(\frac{tp - te}{6} \right)^2$$





$$\text{Expected projected duration} = 7 + 14 + 11 + 4 \\ = 36$$

$$\text{Project length variance} = \sigma^2 = 4 + 16 + 4 + 1 \\ = 25$$

$$\sigma = \sqrt{25}$$

$$= 5$$

$$Z = \frac{TS - TE}{\sigma} = \frac{40 - 36}{5} = 0.8$$

$$\therefore P(Z \leq 0.8) = 0.7881 = 78.81\%$$