UNIT 3

Unit 3: Activity Planning & Risk Management (Q1 & Q2)

Rank	Question Topic	Occurrences	Max Marks	Paper References
1	**CPM vs. PERT & Project Scheduling:** Explain differences between CPM and PERT; List scheduling techniques; Formulate network model.	5	10 [1]	, [2], [3], [4], [5]
2	**Network Models & Calculations:** Explain Network Diagrams/Models; Define Predecessor/Successor/Parallel activities; Explain Forward Pass & Backward Pass with examples.	5	9 [6]	, [2], [7], [8], [9], [10]
3	**Objectives of Activity Planning:** Explain objectives of activity planning in detail with examples; Sequencing and scheduling steps.	4	10 [2]	, [3], [4], [11]
4	**Risk Management Framework & Process:** Describe IT Project Risk Identification Framework; Explain types of risk; Risk Management Process; Risk Response Strategies.	4	10 [6]	, [12], [11], [9]
5	**Schedule Risk Evaluation:** Explain how to evaluate the risk to the schedule (using PERT).	2	9 [11]	, [10]

Q1) Explain the difference between the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) Formulate a network model for a simple project using either CPM or PERT, and calculate the early start, early finish, late st art, and late finish dates for each activity.[8]

The Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) are both network -based project management techniques, but they differ significantly in their approach and application.

Feature	Critical Path Method (CPM)	Program Evaluation and Review Technique (PERT)
Nature of Model	CPM is a **deterministic** model [1]. It assumes that duration estimates are certain and will not vary [2].	PERT is a **probabilistic** model [1]. It accounts for uncertainty in activity durations [3].
Time Estimates	Uses a **single estimate** for activity duration [1], [2].	Uses **three time estimates**: Optimistic (t_o), Most Likely (t_m), and Pessimistic (t_p) to calculate a weighted average duration [1], [4].
Focus	Focuses on the trade-off between **cost and time**, emphasizing cost optimization [1], [2].	Focuses primarily on **time targets** and meeting scheduled deadlines [1].
Application	Suitable for repetitive or routine projects where time estimates are well-known, such as **construction or maintenance** [1], [2].	Suitable for non-repetitive, complex projects like **Research & Development (R&D)** where time cannot be easily predicted [1].
Critical Path	The critical path is calculated based on the longest path using fixed durations [2].	The critical path is determined using the expected time (t_e) derived from the three estimates [3].

Formulating a Network Model and Calculating Schedule Dates

To formulate a network model, project activities are represented as nodes (in Activity-on-Node) or arrows (in Activity-on-Arrow), linked to show dependencies,. The calculation of schedule dates involves two major phases: the **Forward Pass** and the **Backward Pass**.

1. The Forward Pass (Determining Early Start & Early Finish)

The forward pass calculates the earliest dates an activity can begin and end without delaying the project.

- **Early Start (ES):** The earliest time an activity can start.
  - For the first activity, $ES = 0$.
  - For subsequent activities, the ES is determined by the **maximum** Early Finish (EF) of its predecessor activities.
  - *Rule:* If an activity has multiple predecessors (merge point), $ES = \text{Max}(EF \text{ of all predecessors})$.
- **Early Finish (EF):** The earliest time an activity can finish.
  - **Formula:** $EF = ES + \text{Duration}$.

2. The Backward Pass (Determining Late Start & Late Finish)

The backward pass calculates the latest dates an activity can start and finish without delaying the project completion date.

- **Late Finish (LF):** The latest time an activity can finish without delaying the project.
  - For the last activity, LF equals the project completion time (its EF).
  - For other activities, LF is determined by the **minimum** Late Start (LS) of its successor activities.
  - *Rule:* If an activity has multiple successors (burst point), $LF = \text{Min}(LS \text{ of all successors})$.
- **Late Start (LS):** The latest time an activity can start without delaying the project.
  - **Formula:** $LS = LF - \text{Duration}$.

3. Illustrative Example

Consider a simple project with the following activities and durations:

- **Activity A:** Duration = 5 days (Start Node)
- **Activity B:** Duration = 3 days (Predecessor: A)
- **Activity C:** Duration = 2 days (Predecessor: B)

Step 1: Forward Pass (Calculating ES and EF)

1. **Activity A:**
   - $ES = 0$ (Project Start)
   - $EF = 0 + 5 = 5$
2. **Activity B:**
   - $ES = 5$ (Predecessor A's EF)
   - $EF = 5 + 3 = 8$
3. **Activity C:**
   - $ES = 8$ (Predecessor B's EF)
   - $EF = 8 + 2 = 10$ (Project Completion Time)

Step 2: Backward Pass (Calculating LF and LS)

1. **Activity C:**
   - $LF = 10$ (Project Completion Time)
   - $LS = 10 - 2 = 8$
2. **Activity B:**
   - $LF = 8$ (Successor C's LS)
   - $LS = 8 - 3 = 5$
3. **Activity A:**
   - $LF = 5$ (Successor B's LS)
   - $LS = 5 - 5 = 0$

Step 3: Critical Path Identification The critical path is the sequence of activities where the **Float** (or Slack) is zero 7.

- **Float Formula:** $\text{Float} = LS - ES$ or $LF - EF$ 7.
- In this example, for Activity A, B, and C, $LS - ES = 0$. Therefore, the critical path is **A → B → C** 7.

Q2) List different project scheduling Techniques?

1. Critical Path Method (CPM)

• **Description:** CPM is a deterministic model used for projects with predictable activities and well-known duration estimates. It focuses on identifying the longest path of dependent activities (the critical path) which determines the shortest possible ti me to complete the project.

• **Key Feature:** It uses a single time estimate for each activity and emphasizes the trade-off between cost and time.

2. Program Evaluation and Review Technique (PERT)

• **Description:** PERT is a probabilistic model designed for projects with high uncertainty, such as Research & Development (R&D). It accounts for variations in activity durations.

• **Key Feature:** It uses three time estimates for each activity to calculate a weighted average duration: **Optimistic (*to*), Most Likely (*tm*)**, and **Pessimistic (*tp*)**.

3. Gantt Charts (Bar Charts)

• **Description:** This is one of the simplest and oldest techniques for tracking project progress. It represents the project schedule visually with a horizontal bar chart where the length of the bar corresponds to the duration of the activity.

• **Key Feature:** It provides an immediate visual indication of activity start and end dates, but does not explicitly show the dependencies between activities as effectively as network diagrams.

4. Precedence Diagram Method (PDM)

• **Description:** Also known as Activity-on-Node (AON), this method represents activities as nodes and connects them with arrows to show dependencies. It is the basis for modern project management software.

• **Key Feature:** It supports different types of dependencies (Finish-to-Start, Start-to-Start, etc.) and avoids the need for "dummy" activities often required in Activity-on-Arrow networks.

5. Critical Chain Project Management (CCPM)

• **Description:** A scheduling method that focuses on resource availability and managing buffers rather than just task deadlines. It addresses the tendency for work to expand to fill the time available (Parkinson's Law).

• **Key Feature:** It introduces the concept of **Project Buffers** (at the end of the project) and **Feeding Buffers** (where non-critical paths feed into the critical chain) to protect the schedule from uncertainty.

---

q3) How do Network Diagrams help in Project Planning? Define Predecessor, **Successor, and Parallel Activities. Give a real-world example of each.[9]**

How Network Diagrams Help in Project Planning

Network diagrams are fundamental tools in project planning that visualize the project activities and their interrelationships as a graph. They assist the project manager in the following ways:

1. **Sequencing and Dependency Visualization:** They clearly display the logical sequence of tasks, showing which activities must be completed before others can begin. This ensures that the technical constraints and logical flow of the project are respected.
2. **Scheduling (Time Management):** By performing a **Forward Pass** and **Backward Pass** on the network diagram, managers can calculate the **Early Start**, **Early Finish**, **Late Start**, and **Late Finish** dates for every activity.
3. **Identifying the Critical Path:** Network diagrams help identify the **Critical Path**—the longest path through the network that determines the shortest time in which the project can be completed. Any delay on this path will delay the entire project.
4. **Float/Slack Calculation:** They allow for the calculation of **Float** (or Slack), which is the amount of time an activity can be delayed without affecting the project completion date. This helps managers identify which tasks have flexibility and which do not.

5. **Resource Allocation:** By visualizing parallel activities, managers can better plan resource usage to avoid conflicts where two activities compete for the same resources simultaneously.

Definitions and Real-World Examples

To illustrate these concepts, consider a **Construction Project** (specifically building a roof structure) as described in your sources, or a **Software Maintenance Project**.

1. Predecessor Activity

- **Definition:** A predecessor is an activity that must be completed before a subsequent activity can begin. In a network diagram (Activity-on-Node), arrows flow *from* the predecessor *to* the successor.
- **Real-World Example:** In a construction project, **"Lay foundation"** is a predecessor to building the walls or the roof. You cannot physically install the roof until the foundation and frame are in place. Similarly, in software, **"Coding"** is a predecessor to **"Program Testing"**; you cannot test code that hasn't been written.

2. Successor Activity

- **Definition:** A successor is an activity that cannot start until one or more previous activities (predecessors) have been finished.
- **Real-World Example:** In the construction example, **"Decking on roof"** is a successor to **"Frame walls"**. It waits for the framing to be done. In software, **"Installation"** is a successor to **"Program Test"**.

3. Parallel Activities

- **Definition:** Parallel (or concurrent) activities are independent tasks that can be executed simultaneously to shorten the project duration. They occur on different branches of the network diagram that split and later merge.
- **Real-World Example:** In the construction project, **"Assemble roof trusses"** and **"Frame walls"** can be done in parallel. While one team builds the walls on-site, another team can assemble the roof trusses on the ground or off-site, saving time. In a software project, **"Coding"** a module and **"Data take-on"** (preparing the database data) can often happen in parallel.

---

q4) What is network Model? Explain with neat sketch. [9] b) Explain with suitable example forward pass and backward pass. [9]

A **Network Model** (or Network Diagram) is a schematic display of the logical relationships among, or sequencing of, project activities. It is a graph where the activities and their interrelationships are represented to visualize the project flow. Th is model serves as the foundation for critical project management techniques such as the **Critical Path Method (CPM)** and the **Program Evaluation and Review Technique (PERT)**.

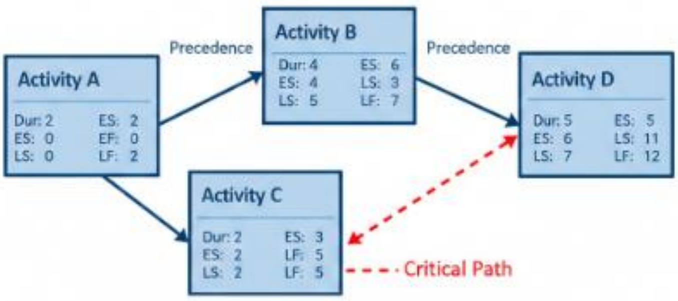
Key Characteristics:

- **Representation:** Activities are typically represented as nodes (in Activity-on-Node or AON) or arrows (in Activity-on-Arrow or AOA).
- **Dependencies:** The lines or arrows connecting these nodes represent the logical dependencies, showing which tasks must precede or follow others.
- **Flow:** The network logic flows from left to right, starting from the initial project event and ending at the project completion.

Sketch Description (Activity-on-Node) While I cannot draw a file directly, the sources describe the standard "Activity-on-Node" (AON) sketch structure used in these models:

1. **Nodes (Boxes/Circles):** Represent the **Activities**. Each node contains data fields for the Activity Name, Duration, Early Start (ES), Early Finish (EF), Late Start (LS), and Late Finish (LF),.

2. **Arrows:** Represent the **Precedence** relationships. An arrow pointing from Activity A to Activity B indicates that A is a predecessor to B,.



Structure of a Node in a Network Model: According to standard labelling conventions in network models, a node is often divided into cells containing the following:

Earliest Start (ES)	Duration	Earliest Finish (EF)
Activity Name	**Description**	**Activity ID**
Latest Start (LS)	**Float**	**Latest Finish (LF)**

b) Explain with suitable example forward pass and backward pass.

The calculation of the project schedule involves two passes through the network diagram: the **Forward Pass** and the **Backward Pass**.

1. The Forward Pass

The forward pass is carried out to calculate the **earliest dates** on which each activity can be started and completed 4 , 1 .

- **Objective:** To determine the project completion time and the Early Start (ES) and Early Finish (EF) for every task 1 .
- **Calculation Logic:**
  - **Early Start (ES):** For the very first activity, $ES = 0$. For subsequent activities, the ES is the maximum Early Finish (EF) of its predecessors 1 , 6 .
  - **Early Finish (EF):** Calculated as $EF = ES + \text{Duration}$ 1 .
- **Merge Rule:** If an activity has multiple predecessors (paths merge), the **Late Finish of the predecessors** determines the start, but specifically, you must choose the **largest (maximum)** EF value from the preceding activities to ensure all prerequisites are done 7 .

2. The Backward Pass

The backward pass enables the calculation of the **latest dates** an activity can start and finish without delaying the overall project end date 5 , 1 .

- **Objective:** To calculate the Late Start (LS), Late Finish (LF), and the **Float** (slack) 5 .
- **Calculation Logic:**
  - **Late Finish (LF):** For the last activity, the LF is equal to the project's Early Finish (determined in the forward pass). For other activities, the LF is determined by the Late Start (LS) of the succeeding activities 1 .
  - **Late Start (LS):** Calculated as $LS = LF - \text{Duration}$ 5 , 1 .
- **Burst Rule:** If an activity has multiple successors (path splits), the LF is determined by the **smallest (minimum)** LS value of the succeeding activities 5 .

Suitable Example

Consider a simple project fragment with three activities connected in a sequence: **Start** $\rightarrow$ **A** $\rightarrow$ **B** $\rightarrow$ **End**.

- **Activity A:** Duration = 4 weeks.
- **Activity B:** Duration = 3 weeks (Dependent on A).

Step 1: Forward Pass Calculation

1. **Activity A:**
   - **ES:** 0 (Start of project) [7].
   - **EF:** $0 + 4 = 4$ weeks [7], [1].
2. **Activity B:**
   - **ES:** 4 (Taken from A's EF) [7].
   - **EF:** $4 + 3 = 7$ weeks [7], [1].
   - *Result:* The project duration is **7 weeks**.

Step 2: Backward Pass Calculation

1. **Activity B:**
   - **LF:** 7 (The project completion time calculated above) [5].
   - **LS:** $7 - 3 = 4$ weeks [5], [1].
2. **Activity A:**
   - **LF:** 4 (Taken from B's LS) [5].
   - **LS:** $4 - 4 = 0$ weeks [5], [1].

Outcome: By comparing the results of the forward and backward passes, we can calculate the **Float** (or Slack).

- **Formula:** $\text{Float} = LF - EF$ or $LS - ES$ [8].
- In this example, for Activity A: $4 - 4 = 0$. Since the float is zero, Activity A is on the **Critical Path** [8].

---

q5) **Explain objectives of activity planning in detail with suitable example.[9]**

Introduction to Activity Planning

Activity planning involves estimating the efforts required for a project—both for the project as a whole and for individual activities—and designing a project plan. The result is a project schedule that indicates the start and completion times for each activity. A plan must be expressed as a collection of objectives so that success or failure can be clearly measured.

Objectives of Activity Planning

The primary objectives of activity planning are to ensure the project is executed efficiently and effectively. These objectives are categorized as follows:

1. Feasibility Assessment

- **Objective:** To determine if the project is realistic and achievable.
- **Explanation:** A feasibility study is an assessment of the practicality of a proposed plan. Activity planning helps assess whether the project targets (such as the deadline and budget) can be met given the available resources and constraints. It ensures that the project timeline adheres to the required timescales and resource constraints.
- **Key Question:** *Is it worth solving? Can we actually deliver this on time?*

2. Resource Allocation

- **Objective:** To manage and assign scarce resources efficiently.
- **Explanation:** Resources (people, hardware, software licenses) are usually limited. The project manager must determine the appropriate time to allocate these resources to specific tasks.
- **Benefits:**
  - Ensures resources are available exactly when needed.
  - Avoids **resource contention**, where two different activities compete for the same resource at the same time.
  - Optimizes the cost by ensuring resources are not sitting idle.

3. Detailed Costing

- **Objective:** To produce accurate financial estimates.
- **Explanation:** A detailed activity plan allows for precise cost estimation. Once activities are scheduled and resources are

allocated, the project manager can calculate the cost of staff hours, equipment time, and overheads for each specific task. This helps in creating a detailed budget and a forecast for cash flow,.

4. Motivation

- **Objective:** To provide targets and goals for the project team.
- **Explanation:** A plan provides clear targets. Staff motivation is often improved when they have specific, achievable goals (like a deadline for a module completion) rather than vague instructions. Meeting these intermediate targets (milestones) provides a sense of achievement and progress.

5. Coordination

- **Objective:** To facilitate communication and teamwork.
- **Explanation:** The project plan serves as a communication tool. It helps coordinate the efforts of different teams (e.g., developers, testers, designers), ensuring that everyone knows what they need to do and when. It is particularly crucial for l arge projects involving multiple project teams.

Suitable Example: "University Exam Management System" Project

Imagine a project to build a web-based **Exam Management System** for a university. Here is how activity planning objectives apply:

1. **Feasibility Assessment:**
    - The university demands the system be live before the **May 1st exam cycle**. Activity planning breaks down the work (Requirements -> Design -> Code -> Test) to see if it is mathematically possible to finish by April 30th. If the plan shows completion in June, the project is **not feasible** under current constraints, and the plan must be adjusted (e.g., removing features or adding staff).
2. **Resource Allocation:**
    - There is only **one Senior Database Administrator (DBA)**. The activity plan ensures that the "Database Design" task is scheduled *before* the "Backend API Development" starts. It also ensures the DBA is not double-booked to work on an unrelated "Library System" project during the critical design week.
3. **Detailed Costing:**
    - The plan shows that the "Testing Phase" requires 3 QA engineers for 4 weeks.
    - *Calculation:* 3 QA × 160 hours × $50/hour = $24,000. This detailed cost is added to the total project budget.
4. **Motivation:**
    - The project manager sets a milestone: **"User Login Module Complete by Feb 15th."** The development team focuses their energy to hit this specific date, knowing it is a measure of their success.
5. **Coordination:**
    - The "Frontend Team" cannot build the student dashboard until the "API Team" finishes the data endpoints. The activity plan makes this dependency visible (Finish-to-Start), so the Frontend Team knows exactly when they can start their work, preventing delays and confusion.

---

q6) discuss the different types of project schedules. Describe the steps involved in sequencing and Scheduling activities.

Types of Project Schedules

Based on the provided texts, project schedules can be categorized by their visual representation, their calculation method, o r the type of calendar system used.

1. Gantt Charts (Bar Charts) This is the first and most common type of project schedule development tool. It represents the project schedule visually with a horizontal bar chart where the length of the bar corresponds to the duration of the activity .

- It is useful for viewing the planned project activity versus the actual activity.
- It serves as a visual cue for tracking project advancement.

2. Network Planning Models (CPM & PERT) These are schematic displays of the logical relationships (dependencies) among project activities.

- **Critical Path Method (CPM):** A deterministic model used for projects where activity durations are known and predictable. It focuses on the trade-off between cost and time.
- **Program Evaluation and Review Technique (PERT):** A probabilistic model used when there is uncertainty in activity durations. It uses three time estimates (Optimistic, Most Likely, Pessimistic) to cal culate expected duration,.

3. Project Calendars (Classic vs. Smart) Project management software often categorizes schedules based on the calendar system used to track tasks:
• **Classic Calendars:** These are hand-made by the user. They offer a flexible option that enables you to simply present the tasks that you have specifically chosen. You can manually create different views (e.g., monthly, quarterly).
• **Smart Calendars:** These are essentially a specific project folder that displays things that meet specific requirements (e.g., tasks assigned to a specific team member). They utilize filters to automatically populate the schedule based on criteria like "start date" or "assigned to".

Steps Involved in Sequencing and Scheduling Activities
The process of sequencing and scheduling transforms a list of requirements into a timeline. According to the "Project Scheduling Process" outlined in the sources, the steps are as follows:
1. Identify Activities The first step is to break down the software requirements into specific, manageable tasks or activities.
• This involves dividing the project into tasks and estimating the duration and resources required to complete each task.
• A **Work Breakdown Structure (WBS)** is typically used to list these activities in a structured manner.

2. Identify Dependencies (Sequencing) Once activities are identified, their logical relationship must be determined. This is the **sequencing** phase,
• The project manager must arrange the tasks so that no task is delayed waiting for another to complete unless necessary.
• This involves identifying which tasks are **predecessors** (must happen before), **successors** (must happen after), or **parallel** (can happen simultaneously).

3. Estimate Resources For each identified activity, the project manager must estimate the resources required. This includes calculating the effort and time needed for execution.

4. Allocate People (Resource Allocation) This step involves assigning specific staff members or teams to the identified activities.
• The schedule must account for the **availability of staff**. It involves allocating specific resources to tasks while ensuring there are no conflicts (e.g., two tasks competing for the same person at the same time),.

5. Create Project Charts Finally, the information is synthesized into project charts (such as Activity charts or Bar charts).
• This produces a detailed plan against which actual achievement can be measured.
• The resulting schedule indicates the **start and stop** times for each activity and clearly shows which staff does which activity.
----
q7) a) Describe IT Project Risk Identification Framework. Explain the types of risk with examples. **[9]**

IT Project Risk Identification Framework
The IT Project Risk Identification Framework is a systematic process designed to deal with risks throughout a project's lifecycle. It is not a one-time activity but a continuous cycle that ensures potential threats are recognized, evaluated, and managed effectively.

Based on the provided sources, the framework typically follows these key stages,:
1. **Risk Identification:** This is the first and arguably most critical step. It involves discovering possible risks that could affect the project.

- **Goal:** To generate a list of potential risks.
- **Techniques:** Common methods include brainstorming, checklists, interviewing stakeholders, and using SWOT analysis.
- **Approaches:**
  - **Checklists:** Using lists of risks found in previous software development projects.
  - **Brainstorming:** Gathering the team to collaboratively identify potential hazards.
  - **Taxonomy-based:** Breaking down possible risk sources using a knowledge base.

2. **Risk Analysis (Assessment):** Once risks are identified, they are analyzed to understand their nature.

- **Goal:** To determine the likelihood (probability) of the risk occurring and the potential impact (damage) if it does,.
- **Outcome:** Risks are often prioritized (e.g., High, Medium, Low) to focus resources on the most critical threats.

3. **Risk Planning:** This stage involves developing strategies to deal with the identified risks.

   ◦ **Goal:** To formulate response plans.

   ◦ **Strategies:** Common strategies include **Risk Acceptance** (doing nothing), **Risk Avoidance** (changing plans to prevent the risk), **Risk Transfer** (shifting risk to a third party, e.g., insurance), and **Risk Mitigation** (reducing impact or likelihood),.

4. **Risk Monitoring:**

   ◦ **Goal:** To track identified risks, monitor residual risks, identify new risks, and ensure risk plans are executed effectively throughout the project lifecycle.

----

b) Types of Risks with Examples

Risks in IT projects are generally categorized into three main types based on what they affect: **Project Risks**, **Technical (Product) Risks**, and **Business Risks**. Additionally, they can be classified by their predictability,,.

1. Project Risks

These risks threaten the project plan itself. They affect the project's schedule, budget, or resources. If these risks become reality, the project schedules may slip and costs may increase.

• **Focus:** Resources, Budget, Schedule.

• **Examples:**

   ◦ **Staff Turnover:** Key personnel leaving the project midway, leading to delays (also known as Personnel Shortfalls).

   ◦ **Unrealistic Schedules:** Management setting deadlines that are impossible to meet with the current resources.

   ◦ **Budget Risks:** The project running out of money before completion due to poor estimation.

2. Technical (Product) Risks

These risks threaten the quality and timeliness of the software to be produced. They relate to the implementation difficulty, design problems, or technical uncertainty.

• **Focus:** Design, Implementation, Interfacing, Verification, Maintenance.

• **Examples:**

   ◦ **Complex Technology:** The team may struggle to use a new, unproven technology or framework effectively.

   ◦ **Performance Failure:** The software functions correctly but fails to meet performance requirements (e.g., it crashes under heavy user load).

   ◦ **Ambiguous Requirements:** The requirements are constantly changing (Gold plating) or were never clearly defined, leading to "feature creep".

3. Business Risks

These risks threaten the viability of the software product. Even if the project is finished on time and the product works technically, it may still fail in the market.

• **Focus:** Marketability, Sales, Management Support.

• **Examples:**

   ◦ **Market Risk:** Building a product that no one actually wants or needs (no market demand).

   ◦ **Strategic Risk:** The product no longer fits the overall business strategy of the company.

   ◦ **Competitive Risk:** A competitor releases a superior or cheaper product before your project is completed, rendering your product obsolete.

4. Predictable vs. Unpredictable Risks

• **Predictable Risks:** Risks that can be anticipated based on past experience (e.g., staff turnover is common).

• **Unpredictable Risks:** Risks that are extremely difficult to identify in advance (e.g., a sudden change in government regulations or a natural disaster).

Q8) Define risk management and explain the different stages of the risk management process. Discuss the different techniques for risk identification and prioritization.

Risk management is the process of minimizing potential issues that can adversely affect a project's schedule. It involves identifying, analyzing, and responding to project risks to maximize the results of positive events and minimize the consequences of adverse events. The goal of risk management is to balance risk and reward, managing feasibility throughout the project.

Stages of the Risk Management Process

The risk management process is a continuous cycle designed to deal with uncertainties. The primary stages are:

1. **Risk Identification:** This is the first step, where the project team identifies possible threats that could impact the project. It involves generating a list of potential risks, such as personnel shortfalls or unrealistic schedules.
2. **Risk Analysis (Assessment):** Once identified, risks are assessed to determine their significance. This involves evaluating the **likelihood** (probability) of the risk occurring and the potential **impact** (damage) if it does. This stage helps distinguish between tolerable risks and those that require immediate attention.
3. **Risk Planning:** In this stage, strategies are formulated to handle the identified risks. Common strategies include **Risk Acceptance** (doing nothing), **Risk Avoidance** (changing plans to prevent the risk), **Risk Reduction/Mitigation** (taking steps to lower probability or impact), and **Risk Transfer** (shifting the risk to a third party).
4. **Risk Monitoring:** This involves tracking identified risks, monitoring residual risks, identifying new risks, and ensuring that risk management plans are executed effectively throughout the project lifecycle.

Techniques for Risk Identification and Prioritization

1. Risk Identification Techniques

Project managers use various methods to discover risks:

- **Checklists:** This method uses lists of hazards and risks discovered in previous software development projects. It is a way to ensure that common pitfalls, such as "personnel shortfalls" or "gold plating," are not overlooked.
- **Brainstorming:** The project team and stakeholders gather to collaboratively identify potential hazards. This leverages the collective knowledge and creativity of the group to find risks that might be unique to the specific project.
- **Taxonomy-Based:** This involves breaking down possible risk sources using a classification system or knowledge base to systematically identify risks.
- **Scenario-Based:** The team creates scenarios to visualize what could go wrong and identifies risks based on those hypothetical situations.

2. Risk Prioritization Techniques

Since resources are limited, risks must be prioritized to focus on the most critical threats:

- **Risk Matrix:** Risks are mapped onto a matrix based on their **Probability** and **Impact**. This visual representation helps categorize risks as High, Medium, or Low priority.
- **Risk Exposure Calculation:** This is a quantitative technique where risk is prioritized by calculating **Risk Exposure = (Potential Damage) × (Probability of Occurrence)**. This provides a numerical value to rank risks.
- **Tolerable vs. Intolerable Risks:** Prioritization involves determining if a risk is "tolerable" (negligible impact or low likelihood) or if it poses a significant threat to the budget or schedule, requiring immediate action.

Q9) Explain the different risk response strategies and how to evaluate the risk to the schedule.

1. Risk Response Strategies

Once risks have been identified and prioritized, the project manager must decide how to handle them. Strategies are generally categorized based on whether the risk is a **negative threat** or a **positive opportunity**.

A. Negative Risk Strategies (Threats)

These strategies aim to minimize the impact of adverse events,.

1. **Risk Avoidance:**
    a. This involves changing the project plan to eliminate the specific risk or condition entirely.

   b. *Example:* Using a proven technology instead of a cutting-edge, unstable one to avoid technical failure, or clarifying requirements to avoid ambiguity,.
2. **Risk Mitigation (or Reduction):**
   a. This strategy focuses on lowering the **probability** of the risk occurring or reducing its **impact** if it does occur.
   b. *Example:* Implementing strict quality control testing to reduce the likelihood of bugs, or having a backup server to reduce the impact of hardware failure,.
3. **Risk Transfer:**
   a. The risk is shifted to a third party, usually through insurance, warranties, or outsourcing. This does not eliminate the risk but assigns the responsibility for managing it to someone else.
   b. *Example:* Buying insurance for equipment theft or outsourcing a complex module to a specialist vendor (Fixed Price Contract),.
4. **Risk Acceptance:**
   a. The project team decides to acknowledge the risk but takes no proactive action because the cost of mitigation outweighs the potential loss, or the risk is deemed "tolerable."
   b. *Example:* Accepting that a team member might get sick for a day or two and deciding to handle it only if it happens,.

B. Positive Risk Strategies (Opportunities)

These strategies aim to maximize the benefits of positive events.

1. **Exploit:** Ensuring that a positive risk (opportunity) definitely happens by removing uncertainty (e.g., assigning top talent to finish a task early).
2. **Share:** Allocating ownership of the opportunity to a third party who is better able to capture the benefit (e.g., forming a joint venture).
3. **Enhance:** Increasing the probability and/or positive impact of an opportunity.
4. **Accept:** Being willing to take advantage of the opportunity if it arises, without actively pursuing it.

2. Evaluating Risk to the Schedule

Evaluating the risk to the schedule involves assessing how uncertainties in activity durations can impact the project's completion date. The sources highlight **PERT (Program Evaluation and Review Technique)** as the primary method for this evaluation, alongside **Monte Carlo Simulation.**

A. Using PERT to Evaluate Schedule Risk

PERT is specifically designed to handle uncertainty in activity durations. Unlike CPM (Critical Path Method), which uses a singletime estimate, PERT uses three estimates to calculate a weighted average and assess the probability of meeting deadlines.

The Process:

1. **Three Time Estimates:** For each activity, three durations are estimated:
   - **Optimistic time (a or t_o):** The shortest possible time if everything goes perfectly.
   - **Most likely time (m or t_m):** The time required under normal conditions.
   - **Pessimistic time (b or t_p):** The longest time required if things go wrong, [6].
2. **Calculate Expected Duration (t_e):** A weighted average is calculated using the formula:

$$t_e = \frac{a + 4m + b}{6}$$

   This creates a single duration that accounts for risk, [6].

3. **Calculate Standard Deviation (σ):** This measures the uncertainty or "riskiness" of an activity.

$$\sigma = \frac{b - a}{6}$$

   A higher standard deviation indicates higher risk.

4. **Probability Assessment (z-value):** Using the expected duration and standard deviation of the Critical Path, PERT allows managers to calculate the statistical probability of finishing the project by a specific target date (using the z-score formula and normal distribution tables) [5].

B. Monte Carlo Simulation

This is a more advanced computerized technique.

- It involves running the project schedule thousands of times, randomly selecting activity durations from a probability distribution (e.g., between the optimistic and pessimistic estimates) for each run.
- **Result:** It produces a probability distribution for the total project duration, showing, for example, that there is a "70% chance the project will finish by June 1st".

C. Critical Path Method (CPM) with Buffers

While standard CPM is deterministic, schedule risk can also be evaluated by identifying the **Critical Path** (the longest path). Any delay on this path directly delays the project. Risk evaluation here involves analyzing "near-critical paths" (paths with very little float) which could become critical if a risk event occurs.

---

q10) Difference between forward pass and backward pass. [9] b) Describe with an example how the effect of risk on project schedule is evaluated using PERT. [9]

The Forward Pass and Backward Pass are the two phases of calculation used in Critical Path Method (CPM) and network analysis to determine project scheduling dates.

Feature	Forward Pass	Backward Pass
Objective	The forward pass is carried out to calculate the **earliest dates** on which each activity can be started and completed [1]. It determines the overall project duration [2].	The backward pass is carried out to calculate the **latest dates** an activity can start and finish without delaying the final project deadline [1]. It identifies the **Float** (or slack) and the Critical Path [3].
Direction	Calculations proceed from the **start** event to the **end** event (left to right in the network diagram) [1], [2].	Calculations proceed from the **end** event back to the **start** event (right to left in the network diagram) [1], [3].
Calculations	Calculates **Early Start (ES)** and **Early Finish (EF)**. Formula: $EF = ES + \mathrm{Duration}$ [1].	Calculates **Late Start (LS)** and **Late Finish (LF)**. Formula: $LS = LF - \mathrm{Duration}$ [1].
Initial Value	The ES of the first activity is typically set to zero [2], [1].	The LF of the last activity is set equal to the project's Early Finish (calculated in the forward pass) or a specified target deadline [1], [3].
Merge/Burst Rules	**Merge Event Rule:** If an activity has multiple predecessors, its ES is the **maximum** EF of all preceding activities [2].	**Burst Event Rule:** If an activity has multiple successors, its LF is the **minimum** LS of all succeeding activities [3].

b) Effect of Risk on Project Schedule Evaluated using PERT

The Program Evaluation and Review Technique (PERT) evaluates the effect of risk (uncertainty) on the schedule by using probabilistic time estimates rather than a single deterministic value [4]. It acknowledges that activity durations can vary, allowing project managers to calculate the statistical probability of meeting deadlines [5], [6].

Process of Evaluating Risk

1. **Three Time Estimates:** Instead of one duration, PERT requires three estimates for each activity to capture uncertainty [7], [8]:
   - **Optimistic time (a or t_o):** The shortest time if everything goes perfectly [7].
   - **Most Likely time (m or t_m):** The time required under normal conditions [7].
   - **Pessimistic time (b or t_p):** The longest time required if things go wrong (risk materializes) [7].
2. **Calculate Expected Duration (t_e):** Risk is incorporated by calculating a weighted average that biases the result toward the "Most Likely" estimate but accounts for the extreme values [8].
   - **Formula:** $t_e = \frac{a+4m+b}{6}$ [7].
3. **Calculate Standard Deviation (σ):** This measures the degree of uncertainty or risk associated with the activity. A larger range between optimistic and pessimistic times results in a higher standard deviation, indicating higher risk [7], [5].
   - **Formula:** $\sigma = \frac{b-a}{6}$ [7].
4. **Probability Assessment:** Using the standard deviation of the Critical Path activities, managers calculate the z-score to determine the percentage probability of finishing by a specific date [5].

Example

Consider a software module development task with the following estimates due to technical uncertainty:

- **Optimistic time (a):** 2 weeks (experienced developer, no bugs).
- **Most Likely time (m):** 5 weeks (normal pace).
- **Pessimistic time (b):** 14 weeks (major bugs found, novice developer).

Step 1: Calculate Expected Duration (t_e)

$$t_e = \frac{2 + 4(5) + 14}{6} = \frac{2 + 20 + 14}{6} = \frac{36}{6} = 6 \text{ weeks}$$

Effect: Although the "most likely" time is 5 weeks, the high risk of the pessimistic scenario pushes the planned duration to 6 weeks [7].

Step 2: Calculate Standard Deviation (σ)

$$\sigma = \frac{14 - 2}{6} = \frac{12}{6} = 2$$

This value (2) represents the risk volatility.

Step 3: Evaluate Schedule Risk If the project manager wants to know the probability of finishing this task in **5 weeks** (the original "most likely" estimate), they use the z-score formula (assuming a normal distribution):

$$z = \frac{\text{Target Time} - \text{Expected Time}}{\sigma} = \frac{5 - 6}{2} = -0.5$$

Using standard probability tables, a z-score of -0.5 indicates approximately a **30.85%** chance of finishing in 5 weeks. This low probability alerts the manager that the risk is high, and the schedule should likely rely on the 6-week estimate or include a buffer [5], [6].

---

q11) With the neat sketch explain formulating a network model.

Formulating a Network Model

A network model (or network diagram) is a schematic display of the logical relationships among project activities. It represents the activities and their interrelationships as a graph, serving as the foundation for scheduling techniques like CPM (Critical Path Method) and PERT.

There are two primary ways to formulate these models:

1. **Activity-on-Arrow (AOA):** Activities are represented by lines (arrows) and events (start/finish) are represented by nodes (circles).
2. **Activity-on-Node (AON) / Precedence Diagram Method (PDM):** Activities are represented as nodes (boxes), and the lines between them show the dependencies,. This is the more common method in modern software project management.

Neat Sketch Representation

```
      [ Start Node ]
            |
            v
   +---------------------+
   | Specify Overall System|
   +----------+----------+
              |
     +--------+--------+
     |                 |
     v                 v
 +---------+      +---------+
 | Specify |      | Specify |
 | Module A|      | Module B|
 +----+----+      +----+----+
      |                |
     +--------+--------+
              |
              v
     +------------------+
     | Check Specification|
     +------------------+
              |
              v
       [ End Node ]
```

Explanation of the Model Formulation

To formulate a network model like the one sketched above, the following steps and rules are applied:

1. identifying Activities (Nodes) In the Activity-on-Node method, each box represents a specific task or activity (e.g., "Specify Module A"). Each node typically contains data such as the Activity Name, Duration, Early Start, and Late Finish dates.

- *Source Concept:* The model displays the sequence of activities. The start and end of the activity are clearly defined.

2. Defining Dependencies (Arrows) The lines or arrows connecting the nodes represent the **precedence** relationships.
- **Predecessor:** An activity that must be completed before another can begin. In the sketch, "Specify Overall System" is a predecessor to "Specify Module A".
- **Successor:** An activity that follows another. "Check Specification" is a successor to both Module A and Module B.
- **Parallel Activities:** The sketch shows "Specify Module A" and "Specify Module B" occurring on different branches simultaneously; these are parallel activities.

3. Flow and Direction
- **Left-to-Right Flow:** Time flows from left to right (or top to bottom in some hierarchies). The network logic starts from the initial project event and moves toward project completion.
- **No Loops:** A network may not contain loops (e.g., Activity A leads to B, which leads back to A). The path must move forward in time.

4. Merge and Burst Points
- **Burst Point:** A node where a single path splits into multiple paths (e.g., after "Specify Overall System").
- **Merge Point:** A node where multiple paths converge into one (e.g., before "Check Specification"). All preceding activities must finish before the merge activity can start.