1) What are the characteristics that make software projects different from other projects? / Mention the characteristics of software projects.

a) Based on logical work, while others are based on physical work.

b) We can't measure its complexity until we actually work on it.

c) Flexibility; unlike other projects, programmers can modify the software at any stage.

d) It often involves technical complexity such as software development, system integration, and network infrastructure.

e) It often requires special considerations for data security, privacy regulations, etc.

2) Describe the importance of software scope and feasibility analysis in project planning.

a) **Defines Boundaries**: Clarifies what is included in the project, preventing scope creep. **(An example of scope creep is that there are times when clients do not explain their vision properly which may lead to the project manager failing to specify the project scope properly.)**
b) **Sets Objectives**: Establishes clear and achievable project goals.
c) **Aids Resource Allocation**: Helps plan and allocate necessary resources effectively.
d) **Identifies Risks**: Assesses potential risks, allowing for early mitigation strategies.
e) **Enables Budget Estimation**: Facilitates accurate cost estimation and budget planning.

f) **Improves Communication**: Ensures stakeholders have a shared understanding of project goals.

g) **Guides Decisions**: Assists project managers in making informed choices.

h) **Increases Success Rates**: Projects with clear scope and feasibility are more likely to be completed successfully.

3) Write down the four dimensions of software feasibility.

a) **Technical Feasibility**: Can the project be technically achieved?

b) **Economic Feasibility**: Will the project be financially viable?

c) **Legal Feasibility**: Are there legal issues to consider?

d) **Operational Feasibility**: Will the organization be able to implement the project?

4) What are the steps to avoid problems in a project? Explain briefly.

a) **Define Clear Goals**: Establish specific, measurable objectives.

b) **Effective Planning**: Create a detailed project plan with timelines.

c) **Regular Communication**: Keep stakeholders informed and involved.

d) **Risk Management**: Identify and mitigate risks early.

e) **Continuous Monitoring**: Track progress and adjust as needed.

5) Explain the types of contracts and their stages in contract placement in detail.

a) Based on Completed Software Application:

i) Bespoke Software: Creates custom software according to the client's needs. (Q: What is a bespoke system?)

ii) Off-the-shelf Software: Pre-made software that can be purchased and used immediately (e.g., MS Office, Adobe Photoshop).

iii) Customized Off-the-Shelf Software: Falls between bespoke and off-the-shelf. It is first bought off-the shelf and then customized to meet specific business needs.

b) Based on Payment Methods:

i) Fixed Price Contracts

ii) Time and Material Contracts

iii) Fixed Price per Delivered Unit Contracts

6) What is the Significance of a Project Risk Matrix?

A project risk matrix helps identify, assess, and prioritize risks in a project. It visually displays risks based on their likelihood of occurring and their potential impact on the project. This helps teams **focus on the most critical risks** that could affect project success.

Importance:

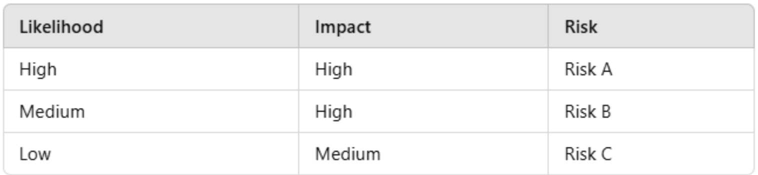
1. Prioritization: It helps identify which risks need immediate attention.
2. Improved Planning: Teams can develop strategies to mitigate or manage risks.
3. Better Communication: It provides a clear overview of risks for stakeholders.

4. Informed Decision-Making: Helps in making better decisions regarding project resources and planning.

Example:

1. Risk A: Heavy rain (High likelihood, High impact)
1. Risk B: Delay in material delivery (Medium likelihood, High impact)
2. Risk C: Minor equipment malfunction (Low likelihood, Medium impact)

In a risk matrix, these risks might be plotted like this:



Likelihood	Impact	Risk
High	High	Risk A
Medium	High	Risk B
Low	Medium	Risk C

In this example, **Risk A** is the most critical and should be addressed first to prevent project delays.

7) With a suitable illustration, explain how risk management is managed during critical parts in software project management.

1. **Identify Risks**: First, identify potential risks that could affect the project's success, such as technical challenges, resource shortages, or changing requirements.
   Example: In a software project, a risk might be that a key developer leaves the team unexpectedly.
2. **Assess Risks**: Evaluate how likely each risk is to happen and what impact it would have on the project.
3. **Plan Responses**: Develop strategies to minimize or eliminate risks. This might include creating backup plans or allocating extra resources.

Example: To manage the risk of a developer leaving, the team might cross-train other members on important tasks.

4. **Monitor Risks**: Continuously track identified risks throughout the project. Look for signs that a risk might occur and be ready to act.
5. **Adjust Plans**: If a risk occurs, adjust the project plan as needed. This may involve reallocating resources or changing timelines.

By effectively managing risks, software projects can stay on track and meet their goals even when challenges arise.

8) Assume there are eight members in a team management setting. Illustrate how decision-making works in team management according to Hackman's Job Characteristics Model, especially since two members are absent due to health issues.

1. **Skill Variety**: The remaining members cover for the absent members by using different skills.
2. **Task Identity**: Focus on completing entire tasks, which helps everyone feel more responsible and proud.
3. **Task Significance**: Understand the importance of their work.
4. **Autonomy**: Members may have more freedom to make decisions and manage tasks better.
5. **Feedback**: Track their progress and make necessary adjustments through regular communication.

9) What is the role of the "team" in decision making? Discuss.

1. **Diverse Perspectives**: Team members bring different viewpoints and ideas, which helps in exploring all options.
2. **Shared Responsibility**: Decisions made by a team are shared, leading to greater commitment and accountability.

3. **Collaborative Problem-Solving**: Teams can work together to find creative solutions to challenges.
4. **Faster Decisions**: A team can make decisions more quickly by pooling knowledge and resources.
5. **Support and Feedback**: Team members provide support and feedback during the decision-making process, ensuring better outcomes.
6. **Common Goal**: The team works towards a common agreement, which increases the chances of successful implementation.

In summary, involving a team in decision-making leads to better ideas and stronger support for the final choice.

10) How does PERT work in activity planning?

PERT (Program Evaluation and Review Technique) helps in planning by breaking down projects into tasks, estimating completion times, and identifying task dependencies. It allows project managers to visualize timelines and track progress effectively.

PERT helps in planning and tracking projects by following these steps:

1. List Activities: Break the project into smaller tasks that need to be done.

2. Set Order: Determine which tasks depend on others before they can start.

3. Estimate Time: For each task, estimate 3 times:

a) Optimistic Time: The shortest time to complete it.
b) Most Likely Time: The usual time it will take.
c) Pessimistic Time: The longest time it might take.

- Calculate the average time using this formula:

$$\text{Expected Time} = \frac{O + 4M + P}{6}$$

4. **Create a PERT Chart:** Draw a diagram showing tasks as points and dependencies as arrows. This helps visualize the project flow.

5. **Find the Critical Path:** Identify the longest path through the chart. This path shows the minimum time to complete the project. Delays here affect the whole project.

6. **Track Progress:** Monitor how long each task takes and compare it to the estimates. This helps spot delays early.

7. **Make Adjustments:** If delays happen, adjust resources or timelines to keep the project on track.

11) How do you prioritize data collection using Earned Value Analysis? Discuss with suitable illustrations.

Earned Value Analysis (EVA) is a project management technique used to measure project performance and progress by comparing planned progress with actual progress. To prioritize data collection, focus on three key metrics:

1. **Planned Value (PV)**: The value of work scheduled by a certain date.
2. **Earned Value (EV)**: The actual value of completed work.
3. **Actual Cost (AC)**:

Example Illustration: Suppose a project's budget is $10,000, and it is supposed to be 50% complete at the end of the first month.

- **Planned Value (PV)**: The scheduled cost for 50% work is $5,000.
- **Earned Value (EV)**: If only 40% of the work is completed, the EV is $4,000.

Suppose the AC is $6,000.

Schedule Variance (SV):

SV = EV - PV

SV = $4,000 - $5,000 = -$1,000 (Delayed behind schedule)

Cost Variance (CV):

CV = EV - AC

CV = $4,000 - $6,000 = -$2,000 (Over budget)

By gathering data on these metrics, you can calculate schedule and cost variances, allowing you to identify delays and budget overruns early.

12) Discuss with a suitable example the process of "selecting the right person for the job" in detail. / How will you ensure that the right person is selected for the job?

Ans: **Job Analysis**: First, understand what the job requires. List the skills, experience, and qualities needed.

Example**: For a software developer position, the job analysis may include programming skills, teamwork, and problem-solving abilities.**

Job Posting: Advertise the job through various channels, like online job boards, company websites, and social media.

Resume Screening: Review resumes to find candidates who meet the job requirements.

Interviews: Conduct interviews to assess candidates' skills, experiences, and fit for the company culture.

***Example*: During an interview for the software developer position, ask about past projects and how they work in a team.**

Testing: Use tests or assessments to evaluate specific skills related to the job.

Reference Checks: Contact previous employers to verify candidates' work history and performance.

Final Selection: Choose the candidate who best fits the job requirements and company culture.

13) What are the three skills for project administrators?

1. **Communication**: Clearly conveying information among team members.
2. **Organizational Skills**: Keeping tasks and resources managed effectively.
3. **Problem-Solving**: Quickly addressing issues that arise during the project.

14) What are Process and Project Metrics?

Process Metrics are used to ***assess the efficiency*** and effectiveness of the software development process. They help organizations understand how well their processes are performing. Common process metrics include:

- **Cycle Time**: The total time taken to complete a specific process, from start to finish.
- **Defect Rate**: The number of defects found in a software product during a specific period, helping teams to gauge the quality of their work.
- **Efficiency**: A measure of the resources used (time, money, effort) versus the output produced, providing insights into how effectively a team is operating.

Project Metrics focus on the performance of the entire project rather than individual processes. They help in tracking the project's health and ensuring it stays on course. Common project metrics include:

- **Budget Variance**: The difference between the budgeted amount and the actual amount spent, indicating if the project is on budget.
- **Schedule Variance**: The difference between the planned completion date and the actual completion date of project tasks, helping teams understand if they are on schedule.
- **Scope Creep**: A measure of how much additional work or changes have been added to the project beyond the original scope.

15) State two of the common software metrics.

a) **Defect Density**: The number of defects per unit of code (e.g., per 1,000 lines).

b) **Code Coverage**: The percentage of code tested by automated tests.

16) What is Size-Oriented Metrics? Explain the pros and cons of it.

Size-oriented metrics measure the size of software products or components, often based on lines of code (LOC) or function points. These metrics help assess productivity, complexity, and overall project size.

Pros of Size-Oriented Metrics:

- **Simplicity**: Easy to understand and calculate, making them accessible for teams.
- **Benchmarking**: Useful for comparing different projects or components based on size.
- **Productivity Measurement**: Helps track the amount of work done in terms of code written or features delivered.

Cons of Size-Oriented Metrics:

- **Not Comprehensive**: Size alone doesn't reflect code quality or functionality.
- **Varied Definitions**: Different teams may measure size in different ways, leading to inconsistencies.
- **Encourages Quantity Over Quality**: Teams might focus on writing more code instead of writing better code.

17) How Metrics Help in Software Projects?

a) **Tracking Progress**: Metrics help monitor the development progress against the project timeline and goals, allowing teams to stay on track.

b) **Identifying Issues**: By analyzing metrics, teams can spot potential problems early, such as code quality issues or schedule delays, enabling proactive management.

c) **Improving Quality**: Metrics related to defects and customer feedback guide teams in enhancing the software quality through informed adjustments.

d) **Facilitating Decision-Making**: Data-driven insights from metrics support better decision-making, helping project managers allocate resources and prioritize tasks effectively.

18) Show the comparison between Measure and Metric.

Measure: A raw data point that quantifies a characteristic (e.g., lines of code, number of bugs).

Metric: A standard of measurement that provides context and can be used for analysis (e.g., defect density, productivity ratio).

19) Why Are Object-Oriented Metrics Better Than Other Metrics?

Object-oriented metrics are often considered better because they focus on the unique features of object-oriented programming. They help measure things like:

- Encapsulation: How well an object keeps its data and methods together. This helps improve code maintainability.
- Inheritance: How classes can share properties and methods, which encourages code reuse.
- Polymorphism: The ability for different classes to be treated as instances of the same class through a common interface, improving flexibility.

These metrics provide a clearer picture of software quality and design, making it easier to identify areas for improvement.

20) Briefly explain Attributes of an Agile Team

An Agile team has several key attributes:

- **Collaboration**: Team members work closely together, sharing knowledge and skills.
- **Flexibility**: They can adapt to changing requirements and priorities quickly.
- **Self-organization**: The team manages its own tasks and responsibilities without needing strict oversight.
- **Continuous improvement**: They regularly reflect on their work to find ways to improve processes and outcomes.

21) Explain the components of reusable software resources.

Reusable software resources include various components that can be used in multiple projects. These components are:

- **Libraries**: Collections of pre-written code that provide specific functionality (e.g., mathematical functions, data manipulation).
- **Frameworks**: Predefined structures that allow developers to build applications faster by providing a foundation for software development.
- **Modules**: Independent pieces of software that can be combined with other modules to create larger systems.
- **APIs (Application Programming Interfaces)**: Sets of rules and protocols that allow different software applications to communicate with each other.

22) Discuss about (i) Organizational structure. (ii) Decision making.

(i) Organizational Structure: Organizational structure refers to how a company arranges its teams and departments to achieve its goals. It defines roles, responsibilities, and the flow of information. Common structures include:

- **Hierarchical**: A clear chain of command with multiple levels of management.
- **Flat**: Few or no levels of middle management, promoting open communication.
- **Matrix**: Employees report to multiple managers for different projects, enhancing flexibility and collaboration.

A well-defined structure helps improve efficiency, communication, and accountability within the organization.

(ii) Decision Making: Decision-making is the process of choosing the best option among alternatives. It involves several steps:

1. **Identifying the problem**: Recognizing the need for a decision.
2. **Gathering information**: Collecting relevant data and insights.

3. **Generating alternatives**: Brainstorming possible solutions.
4. **Evaluating options**: Assessing the pros and cons of each alternative.
5. **Making the choice**: Selecting the best option based on the analysis.
6. **Implementing the decision**: Putting the chosen option into action.
7. **Reviewing the outcome**: Analyzing the results to improve future decisions.

Effective decision-making is crucial for organizational success and can involve individual or group processes, depending on the situation.

23) Discuss the four managerial skills.

a) **Technical Skills**: The ability to understand and apply specific knowledge and techniques related to the job. This is crucial for problem-solving and decision-making.

b) **Interpersonal Skills**: These skills help managers communicate effectively, motivate team members, and resolve conflicts. Good interpersonal skills foster a positive work environment.

c) **Conceptual Skills**: The ability to understand complex situations and see how different parts of the organization work together. This skill helps managers make strategic decisions.

d) **Decision-Making Skills**: The ability to analyze information, evaluate options, and choose the best course of action. Strong decision-making skills are vital for effective leadership and project success.

24) Elaborate the W5HH Principle (5 W's and 1 H)

The W5HH principle helps in analyzing situations by asking six key questions:

1. **Who**: Identify stakeholders and team members involved.

○ *Example*: Who is responsible for the tasks?

2. **What**: Define the issue or project goal.

○ *Example*: What are we trying to achieve?

3. **When**: Determine the timeline and deadlines.

○ *Example*: When is the project due?

4. **Where**: Specify the location or context.

○ *Example*: Where will the project take place?

5. **Why**: Understand the reasons behind the project.

○ *Example*: Why are we doing this?

6. **How**: Explore the methods and processes for execution.

○ *Example*: How will we implement the plan?

Importance

- **Clarity**: Ensures everyone understands the project.
- **Comprehensive Analysis**: Covers all necessary aspects.
- **Improved Decision-Making**: Leads to better-informed choices.
- **Enhanced Collaboration**: Encourages team discussion and input.

25) Which people are involved in Software estimation?

Project Managers: They lead the estimation process, ensuring that estimates align with the project goals, scope, and budget. They coordinate with other team members to gather insights and create realistic estimates.

Developers: Experienced developers provide valuable input based on their technical expertise and understanding of the project requirements. Their knowledge of past projects helps in providing accurate estimates for similar tasks.

Business Analysts: They help clarify project requirements and objectives, making it easier to understand what needs to be done. Their insights into user needs and market trends can also inform the estimation process.

Quality Assurance (QA) Engineers: QA team members can estimate the time and effort required for testing, ensuring that the project estimates include adequate time for quality checks.

26) What is Decomposition technique?

The decomposition technique involves breaking down a large project into smaller, manageable parts or tasks. This helps in estimating time, effort, and resources more accurately.

27) What are the software measurement categories?

Software measurement can be classified into **3 primary categories:**

1. **Product Metrics**: These metrics assess the characteristics of the software product itself, focusing on aspects like size, complexity, and performance. Examples include lines of code, function points, and response time.
2. **Process Metrics**: These metrics evaluate the effectiveness of the software development process. They help organizations identify strengths and weaknesses in their methodologies. Examples include defect resolution time, productivity rates, and process compliance.
3. **Project Metrics**: These metrics focus on overall project performance and progress. They track how well the project is adhering to its schedule and budget. Examples include earned value, schedule variance, and cost performance index.

28) Write down the Mantei's seven factors in team planning.

a) Common Purpose: Clear goals for the team.

b) Roles and Responsibilities: Defined roles for each member.

c) Communication: Open lines of communication.

d) **Decision Making**: Clear processes for making decisions.

e) **Conflict Resolution**: Methods for handling disagreements.

f) **Performance Evaluation**: Regular assessment of team performance.

g) **Continuous Improvement**: Focus on ongoing development.

29) Discuss about the different models of motivation.

1. **Maslow's Hierarchy of Needs**:

○ Needs are ranked like a ladder: basic needs (food, water), safety, social (belonging), esteem (respect), and self-actualization (fulfilling potential). Higher needs are pursued only after lower ones are met.

2. **Herzberg's Two-Factor Theory**:

○ Work factors are divided into hygiene factors (salary, conditions) that prevent dissatisfaction but don't motivate, and motivators (recognition, achievement) that boost satisfaction and performance.

3. **McClelland's Theory of Needs**:

○ People are driven by three needs: achievement (success), affiliation (belonging), and power (influence). The strongest need influences their behavior.

4. **Vroom's Expectancy Theory**:

○ Motivation depends on believing that effort leads to performance and that performance leads to rewards.

○ Formula: Motivation = Expectation × Value of reward.

5. **Adams' Equity Theory**:

○ Motivation is affected by fairness. If people feel treated unfairly compared to others, their motivation decreases.

6. **Skinner's Reinforcement Theory**:

○ Behavior is shaped by rewards and consequences. Rewards encourage good behavior; removing unpleasant things can also motivate change.

7. **Drive Theory**:
   - ○ Proposed by Daniel Pink, this theory highlights three motivators: autonomy (control over work), mastery (improving skills), and purpose (doing meaningful work).
8. **Self-Determination Theory (SDT)**:
   - ○ Motivation comes from three intrinsic needs: autonomy (control), competence (feeling capable), and relatedness (connections with others). Meeting these needs boosts motivation and engagement.

30) What is "Maslow's Hierarchy of Needs"?

Maslow's Hierarchy of Needs is a psychological theory that describes a five-tier model of human needs, often depicted as a pyramid. The levels are:

1. Physiological Needs: Basic survival needs like food and shelter.
2. Safety Needs: Security and protection from harm.
3. Love/Belonging: Social connections and relationships.
4. Esteem: Self-esteem and recognition from others.

Self-Actualization: Achieving personal potential and growth.

31) What do you understand about "Egoless Programming"?

Egoless programming is a philosophy in software development that encourages programmers to set aside personal pride and ego. The idea is to focus on collaboration, **team success**, and constructive feedback **rather than individual ownership of code**. This approach promotes better teamwork and more effective problem-solving.

32) List the important roles of the Configuration Librarian.

a) **Manage Documentation**: Organizes and maintains project documents.

b) **Version Control**: Ensures that the correct versions of files are used.

c) **Track Changes**: Records changes to project items and manages updates.

d) **Coordinate Releases**: Oversees the **release of software versions and updates.**

e) **Facilitate Communication**: Acts as a point of contact for configuration management.

33) Name the popular visual tools used for monitoring and tracking the project progress.

a) **Gantt Charts**: Shows project timelines and tasks.

b) **Kanban Boards**: Visualizes workflow and task status.

c) **Burndown Charts**: Tracks progress against time in Agile projects.

d) **Project Dashboards**: Provides an overview of project metrics and status.

34) What are the factors in the process that influence quality? How does it improve software quality?

a) Team Skills: The expertise of team members impacts the quality of work.

b) Tools and Technology: The effectiveness of tools used affects the outcome.

c) Processes and Standards: Established methodologies ensure consistency and quality.

Improving these factors leads to fewer defects, better functionality, and higher user satisfaction, resulting in overall better software quality.

35) What are the quality factors in a software project?

1. **Functionality**: How well the software meets requirements.
2. **Reliability**: Consistency of performance over time.
3. **Usability**: Ease of use for the end-user.
4. **Efficiency**: Optimal use of resources.
5. **Maintainability**: Ease of updating and fixing the software.

36) What is a Dangle in an Activity Network?

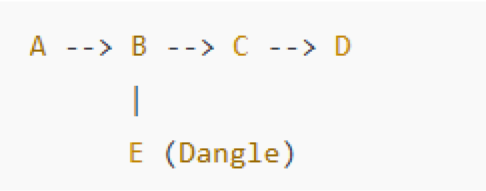
A dangle in an activity network refers to a task or activity that has ***no dependencies or connections to other tasks***. This means that it does not affect or is not affected by any other activities in the project.

Imagine a project with the following tasks:

- A: Start Project
- B: Design Phase
- C: Development Phase
- D: Testing Phase
- E: Dangle Activity (Optional Task)

In this scenario, Task E (the dangle) is not connected to Tasks B, C, or D. It can be completed independently of the other tasks.

Illustration:



```
A --> B --> C --> D
      |
      E (Dangle)
```

In this example, **E** can be done anytime without affecting the sequence of other tasks.

37) What are the 4p's principle in project management?

The 4P's Principle in project management provides a framework for successful project execution. The four elements are:

1. **People:** This refers to the team members involved in the project. Effective communication, collaboration, and clear roles are vital for ensuring that everyone works towards the same goals.
2. **Product:** The product represents the deliverable or outcome of the project. Understanding the product requirements and specifications is crucial for meeting stakeholder expectations.
3. **Process:** This encompasses the methodologies, tools, and practices used to execute the project. Having a well-defined process helps streamline project activities and improves efficiency.
4. **Project:** This refers to the overall plan, including scope, timeline, and resources. Clear project management helps ensure that the project stays on track and achieves its objectives.

38) Briefly explain the ins and outs (key aspects) of a project.

a) **Initiation**: Defining the project's purpose, scope, and objectives.

b) **Planning**: Creating a detailed roadmap that outlines tasks, timelines, and resources needed to complete the project.

c) **Execution**: Carrying out the project plan by coordinating tasks, managing resources, and ensuring team collaboration.

d) **Monitoring and Controlling**: Tracking progress, making adjustments as necessary, and ensuring the project stays on schedule and within budget.

e) **Closure**: Finalizing all project activities, delivering the final product, and reviewing the project's success.

39) Briefly describe the estimation and risk with an example.

Estimation involves predicting the time, cost, and resources needed for a project. For example, if a team estimates that a software feature will take two weeks to complete, they plan their resources and schedule accordingly.

Risk refers to the potential problems that could impact a project. For instance, if the team relies on a third-party API, there's a risk that the API may change or become unavailable, which could delay the project.

Example:

- **Estimation**: A team estimates a project will take 3 months and cost $50,000 based on previous similar projects.
- **Risk**: They identify a risk that a key developer might leave the company, which could affect the timeline. To mitigate this risk, they plan for cross-training team members.

40) How can a process be improved? Explain with examples.

A process can be improved by identifying inefficiencies and implementing changes. For example, if a software development team finds that their code review process takes too long, they might implement automated code review tools to speed up the process.

41) What is cost monitoring? What is its importance?

Cost Monitoring refers to the process of tracking and controlling a project's expenses to ensure they stay within the approved budget.

Importance:

1. Prevents Budget Overruns: Ensures that costs do not exceed the allocated budget.
2. Identifies Cost Variances: Helps detect any deviations between planned and actual costs early on.
3. Improves Financial Planning: Assists in forecasting future expenses and managing resources efficiently.
4. Enhances Decision-Making: Provides accurate financial data for making informed project decisions.
5. Ensures Accountability: Tracks spending and holds team members accountable for financial performance.
6. Supports Project Success: Helps maintain financial control, ensuring the project stays viable and meets financial goals.

42) What is the need for contract management in Software Project Management?

a) **Ensures Clear Expectations**: Defines roles, deliverables, timelines, and costs between parties.

b) **Mitigates Risks**: Protects against legal, financial, and operational risks through well-defined terms.

c) **Ensures Compliance**: Ensures adherence to agreed standards, requirements, and regulations.

d) **Manages Changes**: Facilitates handling of scope changes, cost variations, and disputes.

43) How do you define project activities in software project management?

a) **Break Down Tasks:** Divide the project into manageable tasks (coding, testing, designing).

b) **Sequence Activities:** Arrange tasks in a logical order based on dependencies.

c) **Allocate Resources:** Assign team members, tools, and time for each task.

d) **Set Milestones:** Define key checkpoints for progress and completion.

44) What are the assessments needed in the technical part for Software Project Management?

a) **Feasibility Analysis**: Evaluate if the project is technically possible with available resources.

b) **Risk Assessment**: Identify potential technical risks (integration issues, performance bottlenecks).

45) List any two risk planning and control methods.

a) **Risk Mitigation Plan**: Develop strategies to minimize the impact of risks.

b) **Contingency Planning**: Prepare backup plans in case risks materialize.

46) How do you identify hazards in software Project Management?

a) **Risk Analysis:** Regularly analyze project activities for potential hazards (e.g., security vulnerabilities, technical failures).

b) **Stakeholder Input:** Gather feedback from team members and stakeholders to identify risks early.

47) What factors decide the success of a project?

a) **Clear Objectives**

b) **Effective Planning**

c) **Strong Leadership**

d) **Stakeholder Engagement**

e) **Skilled Team**:

f) **Resource Management**

g) **Risk Management**

h) **Adaptability**:

i) **Monitoring and Control**

j) **Quality Management**: