Q1. What is the difference between Brainstorming and JAD Sessions? 3 Marks

Answer: The difference between **Brainstorming** and **Joint Application Development (JAD) Sessions** lies in their purpose, participants, and structure:

1. **Purpose**:
    - **Brainstorming**: Focuses on generating a large number of creative ideas or solutions for a specific problem. It emphasizes free-flowing thought without immediate evaluation.
    - **JAD Sessions**: Aim to gather detailed requirements and ensure alignment between stakeholders (users, developers, and analysts) during system development.
2. **Participants**:
    - **Brainstorming**: Involves a group of participants, often from diverse backgrounds, who contribute ideas without necessarily being subject-matter experts.
    - **JAD Sessions**: Involve specific stakeholders, including system users, business experts, and IT professionals, to ensure thorough understanding and agreement on requirements.

Aspect	Brainstorming	JAD Sessions
Purpose	Generate a wide range of creative ideas or solutions.	Gather detailed system requirements collaboratively.
Participants	Diverse group, often not subject-matter experts.	Stakeholders including users, analysts, and developers.
Structure	Informal and unstructured, encouraging free-flowing ideas.	Formal and structured with a facilitator and predefined agenda.
Focus	Creativity, innovation, and quantity of ideas.	Collaboration, consensus-building, and precision in requirements.
Output	A list of ideas, potential solutions, or approaches.	Clear and agreed-upon system requirements and decisions.
Facilitation	A moderator may guide but with minimal control.	A trained JAD facilitator ensures discussions stay on track.
Evaluation of Ideas	Ideas are not evaluated immediately to foster creativity.	Ideas and requirements are evaluated and refined during the session.
Use Case	Early stages of problem-solving or project ideation.	System development, requirements gathering, and aligning stakeholders.

Q 2. Why Document Analysis is one of the compulsory technique we use in a Project? Justify – 3 Marks

Answer: Document analysis is essential in a project because it provides **critical insights, context, and clarity** during the requirements-gathering phase and throughout the project lifecycle. Here's why it is indispensable:

1. Identifying Existing Knowledge

- **Understanding the Current State**: Analyzing documents like business process workflows, technical manuals, or existing system documentation helps understand how things are currently done.
- **Building on Existing Work**: Avoids duplication of effort by identifying reusable components or processes.

2. Validating and Refining Requirements

- **Reference for Accuracy**: Documents like contracts, policies, or standards serve as authoritative sources for ensuring requirements are accurate and aligned with business goals.
- **Gap Analysis**: Helps identify missing, incomplete, or conflicting requirements by comparing current documentation with stakeholder input.

3. Ensuring Compliance

- **Regulatory Alignment**: Many industries (e.g., finance, healthcare) have strict compliance and regulatory standards. Document analysis ensures the project adheres to these requirements.
- **Audit Trail**: Provides traceability for decisions by linking requirements to specific regulations, contracts, or guidelines.

4. Supporting Stakeholder Discussions

- **Context for Collaboration**: Documents such as meeting minutes, previous reports, and stakeholder feedback provide context for productive discussions.
- **Clarifying Assumptions**: Prevents miscommunication by uncovering unstated assumptions or undocumented processes.

5. Mitigating Risks

- **Avoiding Misinterpretation**: Relying on verbal communication alone can lead to misunderstandings; written documentation provides a stable reference point.
- **Change Management**: Understanding historical documentation helps anticipate the impact of changes on existing systems or processes.

6. Saving Time and Resources

- **Starting Point for Analysis**: Instead of starting from scratch, document analysis gives a baseline to work from, saving time during requirements elicitation.
- **Fewer Rework Cycles**: Well-documented requirements and processes reduce errors and rework in later stages of the project.

7. Supporting Decision-Making

- **Data-Driven Insights**: Documents like financial reports or usage data help make informed decisions about priorities, scope, and resource allocation.
- **Historical Context**: Past project reports or lessons learned documents inform better strategies for the current project.

When Document Analysis is Most Useful

- **Projects with Existing Systems or Processes**: Provides a foundation to identify improvements or changes needed.
- **Regulated Environments**: Ensures compliance with industry standards or legal requirements.
- **Complex Stakeholder Groups**: Helps consolidate and align different stakeholder perspectives.

Q3. In Which Context we will use Reverse Engineering? - 3 Marks

Answer: Reverse engineering in the context of requirement gathering refers to the process of analyzing an existing system, application, or product to extract its functionalities, behaviors, and requirements. It is used when there is little to no documentation, or when the system needs to be understood for enhancements, integration, or migration.

Reverse engineering is used in various contexts where there is a need to understand, analyze, or recreate an existing system or product. Here are key contexts where reverse engineering is applied:

1. **System's functionality and structure**
   - When working with outdated systems or software where documentation is missing or incomplete, reverse engineering helps understand the system's functionality and structure to support maintenance, upgrades, or migration to newer technologies.
2. **Competitor Analysis**
   - In industries like manufacturing or software, reverse engineering is used to study a competitor's product to understand its design, functionality, or features without violating intellectual property rights.
3. **Troubleshooting and Debugging**
   - Reverse engineering helps identify and resolve issues in complex systems, especially when the original source code or design details are unavailable.

4. **Security Analysis**
    - In cybersecurity, reverse engineering is used to analyze malware, viruses, or security threats to understand their behavior and develop countermeasures.
5. **Recreating Lost Designs**
    - If original design documents or source code are lost, reverse engineering allows the recreation of the product or system for further development or replication.
6. **Integration and Interoperability**
    - When integrating new components into existing systems, reverse engineering helps understand how the old and new components can work together effectively.

Category of reverse engineering :

1. Black Box Approach

Definition:

- The system or component is analyzed, tested, or reverse-engineered **without access to its internal structure, design, or implementation**.
- Focuses only on the **inputs** and **outputs** of the system.

Characteristics:

- No knowledge of internal code, algorithms, or architecture.
- Tests or analysis are conducted from the **user's perspective**.
- Based solely on the system's **functional behavior**.

2. White Box Approach

Definition:

- The system or component is analyzed, tested, or reverse-engineered **with full access to its internal structure, design, and implementation**.
- Focuses on **understanding the internal workings** of the system.

Characteristics:

- Complete visibility of the source code, algorithms, and architecture.
- Tests or analysis are conducted from the **developer's perspective**.
- Covers paths, logic, and internal workflows.

Aspect	**Black Box**	**White Box**

Access to Internals	No access	Full access
Focus	Functional behavior	Internal logic and structure
Perspective	End-user	Developer or system architect
Test Coverage	Limited (input-output based)	Comprehensive (all paths and logic)
Technical Expertise	Minimal	High
Use Cases	Functional testing, usability	Security analysis, performance tuning

In all these contexts, reverse engineering provides valuable insights to support maintenance, innovation, or problem-solving.

Q4. What is the difference between Brainstorming and Focus Groups? - 3 Marks

Answer:

Brainstorming

Brainstorming is a creative problem-solving technique where a group of individuals comes together to generate a wide range of ideas or solutions for a specific problem or goal. It encourages free-thinking, avoids immediate judgment or criticism, and focuses on the quantity of ideas to stimulate creativity and innovation. Brainstorming sessions are often informal and aim to explore possibilities without restrictions.

Focus Groups

A **Focus Group** is a research method where a small, carefully selected group of participants discusses a specific topic, product, or idea under the guidance of a trained facilitator. The goal is to gather qualitative insights, opinions, and feedback to understand behaviors, preferences, or attitudes. Focus groups are structured or semi-structured and are commonly used in market research, product development, or user experience studies.

Aspect	Brainstorming	Focus Groups
Purpose	Generate a wide range of creative ideas or solutions.	Gather opinions, perceptions, and feedback on a specific topic or product.

Participants	Participants can be a mix of individuals, often not predefined, and their expertise may vary.	Typically a selected group of 6–12 participants representing a specific target audience.
Structure	Informal and unstructured, encouraging free-flowing ideas.	Semi-structured or structured with a facilitator guiding discussions.
Focus	Creativity and innovation, with no judgment or evaluation during the session.	Understanding opinions, attitudes, and behaviors in a detailed manner.
Output	A list of ideas or possible solutions.	Insights into user preferences, behaviors, or market trends.
Evaluation	Ideas are not evaluated during the session.	Discussions often involve probing and clarification of participant opinions.
Use Case	Early stages of ideation or problem-solving.	Market research, product feedback, or understanding user needs.

Q5. Observation Technique – Explain both Active and Passive approaches - 3 Marks

Answer: The **Observation Technique** involves studying individuals, processes, or systems to gather information without direct interaction. It helps understand behaviors, workflows, and environments in real-time. There are two primary approaches: **Active Observation** and **Passive Observation**.

Active Observation

- **Definition**: The observer actively engages with the environment or participants, asking questions, taking notes, and sometimes participating in activities.
- **Characteristics**:
    1. Direct interaction with participants.
    2. Observer may influence the environment to gain deeper insights.
    3. Suitable for understanding complex processes or gathering detailed **qualitative** data.
- **Example**: A business analyst shadowing employees in a manufacturing plant and asking clarifying questions about their tasks.

Passive Observation

- **Definition**: The observer remains unobtrusive, simply watching and recording behaviors or processes without interference.
- **Characteristics**:
    1. Non-intrusive and does not affect the natural environment.
    2. Ideal for capturing unbiased, natural behaviors.
    3. Focuses on observing and documenting rather than interacting.

- **Example**: Observing customers in a store to understand purchasing patterns without engaging them in conversation.

Q6. How do you conduct the Requirements Workshop- 3 Marks

Answer: A **Requirements Workshop** is a collaborative meeting where stakeholders come together to define and refine project requirements. Here's how it is conducted:

1. **Planning the Workshop**
    - **Identify Objectives**: Clearly define the purpose of the workshop (e.g., gather requirements, prioritize features).
    - **Select Participants**: Include stakeholders such as business users, project managers, developers, and analysts.
    - **Prepare Agenda**: Create a structured agenda with activities, timelines, and expected outcomes.
    - **Pre-Workshop Preparation**: Share background information, documents, or pre-reading material with participants in advance.

2. **Facilitating the Workshop**
    - **Set Ground Rules**: Establish guidelines for participation, such as respect for ideas and time limits.
    - **Use Facilitation Techniques**: Apply brainstorming, group discussions, or prioritization methods to gather and refine requirements.
    - **Encourage Collaboration**: Promote open communication and active involvement from all participants.
    - **Document Inputs**: Record all ideas, decisions, and requirements in real time using tools like whiteboards, sticky notes, or software.

3. **Post-Workshop Activities**
    - **Validate Outputs**: Review and validate the documented requirements with participants to ensure accuracy.
    - **Create Deliverables**: Summarize the findings into formal documentation, such as a requirements specification or workshop report.
    - **Follow-Up**: Address any unresolved questions or gaps and share the finalized outputs with stakeholders.

Q7. In which context, Interview Technique can be conducted by a BA ? How may approaches are there in conducting Interviews? (Structured – Unstructured) Explain them. Explain the difference between Open Ended Questions and Closed ended Questions – 6Marks

Answer: Business Analysts (BAs) to gather in-depth insights from stakeholders, users, or subject-matter experts regarding business needs, system requirements, or project objectives commonly use the Interview Technique.

Interviews are especially valuable in contexts where direct interaction with stakeholders is needed to clarify, validate, or explore specific information. Below are the contexts in which interviews can be conducted:

- **Requirement Elicitation**: Interviews are used to gather detailed requirements from stakeholders to understand their needs, priorities, and expectations for the system or project.
- **Stakeholder Analysis**: BAs conduct interviews to identify key stakeholders, understand their roles, and assess their interests in the project.
- **Process Improvement**: To understand existing workflows and identify pain points, inefficiencies, or areas of improvement.
- **Risk Identification**: Interviews can help uncover potential risks or challenges by discussing concerns with stakeholders.
- **System Analysis and Design**: Gathering information on how users interact with systems, identifying system limitations, or understanding future needs.
- **Change Management**: Understanding user concerns and attitudes toward new systems or processes.

Approaches to Conducting Interviews

There are two main approaches to conducting interviews: **Structured** and **Unstructured**.

1. Structured Interviews

- **Definition**: A structured interview is a highly controlled and formal approach where the BA follows a predetermined set of questions, usually with limited flexibility. Every participant is asked the same set of questions in the same order.
- **Characteristics**:
    - Predefined questions that are consistent across all interviews.
    - Limited to yes/no or closed-ended questions.
    - Quantitative data collection is more straightforward.
    - Little room for deviation from the script.
- **When to Use**:
    - When gathering specific information from multiple respondents.
    - When the focus is on consistency and comparing responses across different stakeholders.
    - In large-scale surveys or when a high level of uniformity is required.
- **Advantages**:
    - Easier to analyze and compare responses.
    - Reduces bias and keeps interviews consistent.

- **Disadvantages**:
  - Lacks flexibility and may miss important context.
  - Can be rigid and might not allow for probing deeper into issues.

2. Unstructured Interviews

- **Definition**: An unstructured interview is informal and conversational, where the BA explores topics based on the flow of the conversation. There are no predefined questions, and the focus is on gaining insights and exploring ideas in-depth.
- **Characteristics**:
  - Open-ended and flexible questions.
  - The interview may take the shape of a free-flowing conversation.
  - The BA probes and explores different avenues based on the answers provided.
  - Qualitative data is often gathered.
- **When to Use**:
  - When exploring new or complex topics where the BA doesn't know all the aspects in advance.
  - When deeper understanding of the stakeholder’s views, emotions, or motivations is needed.
  - In the early stages of a project to discover hidden issues or needs.
- **Advantages**:
  - Provides rich, in-depth information.
  - Flexible, allowing the BA to explore the answers further.
- **Disadvantages**:
  - Difficult to analyze systematically.
  - Can lead to inconsistent data, making it hard to compare responses.
  - Time-consuming.

Key Differences Between Open-ended and Closed-ended Questions:

Aspect	Open-ended Questions	Closed-ended Questions
Response Type	Detailed, descriptive, and elaborative responses.	Short, direct answers (Yes/No, multiple choice).
Purpose	Explore opinions, experiences, or motivations.	Gather specific facts, quantify opinions, or confirm details.
Data Type	Qualitative data (rich, in-depth).	Quantitative data (easier to analyze statistically).
Time to Answer	Longer, requires thought and explanation.	Quick, minimal explanation required.
Analysis	Difficult to analyze; requires interpretation.	Easy to analyze using statistical methods.
Examples	"How do you feel about the new system?"	"Do you like the new system? (Yes/No)"

Q8. Questionnaire Technique – Where we will use? Give one example - 6 Marks

Answer: The **Questionnaire Technique** is a widely used method for gathering information from a large number of stakeholders, users, or customers. It involves distributing a set of structured or semi-structured questions to respondents, who then provide their answers in writing. Questionnaires are particularly useful when the need arises to collect data from a diverse group in an efficient and standardized manner.

Where We Will Use the Questionnaire Technique?

Questionnaires are typically used in the following scenarios:

1. **Requirement Gathering**:
   - When there is a need to understand the requirements or expectations of a large number of stakeholders, customers, or end-users, a questionnaire can help collect data in a structured way. This is especially useful for identifying common needs and preferences.
2. **Market Research**:
   - Businesses often use questionnaires to understand customer preferences, attitudes, and behaviors. This data can help in shaping product development, marketing strategies, or customer service improvements.
3. **User Feedback and Satisfaction Surveys**:
   - After launching a product or service, questionnaires are used to gather feedback from users regarding their satisfaction, usability, and areas of improvement.
4. **Employee Surveys**:
   - Companies often use questionnaires to gather feedback from employees regarding job satisfaction, work environment, organizational culture, or ideas for improvement.
5. **Process and System Evaluation**:
   - A questionnaire can help assess how well an existing system or process is functioning, identify problems or inefficiencies, and gather input on potential improvements.
6. **Feasibility Studies**:
   - When evaluating the feasibility of a new project or initiative, questionnaires are used to gather information from stakeholders or potential users to assess the viability of the proposed solution.
7. **Regulatory Compliance and Audits**:
   - For compliance-related purposes, questionnaires may be used to assess adherence to legal, financial, or operational standards and regulations.

Example of Using the Questionnaire Technique

Here's an example of a questionnaire you can use for **Requirements Gathering** for the online agriculture product store project. The questions are tailored to understand the stakeholders' needs and expectations:

1. What are the primary goals you want this online store to achieve?
2. Who are the target users of the platform (e.g., farmers, manufacturers)?
3. What challenges are farmers or manufacturers currently facing that this platform should solve?

User Login & Registration

4. What information should users provide during the registration process?
5. Should there be different types of accounts for manufacturers and farmers? If yes, what functionalities should each have?
6. Do you want a two-factor authentication system for account security?

Product Catalog

7. What categories of products (e.g., fertilizers, seeds, pesticides) should be included in the catalog?
8. Should users be able to filter products by brand, price, or other attributes?
9. Is there a preference for how product details (e.g., specifications, usage instructions) should be displayed?

Search Functionality

10. What filters or criteria should users have to refine their search results?
11. Should the search feature support multilingual capabilities for farmers in remote villages?

Payment System

12. What payment methods (e.g., COD, UPI, Debit/Credit cards) are essential?
13. Do you require a payment confirmation email or SMS for users?
14. Should there be an option to save payment methods for faster future transactions?

Order and Delivery

15. Should the platform allow users to track their orders in real-time?
16. What details (e.g., expected delivery date, courier details) should be displayed on the delivery tracking page?
17. Do you want to integrate delivery notifications via email or SMS?

Q9. How to Sort the Requirements – Where we will use? Give one example - 5 Marks

Answer: Sorting requirements is an essential activity in the process of business analysis. It helps to prioritize and organize requirements in a way that ensures the most critical and valuable requirements are addressed first. Sorting can involve categorizing, ranking, or organizing requirements based on factors such as importance, urgency, feasibility, or business value.

Sorting requirements is a process of categorizing and organizing them to streamline their analysis, prioritization, and implementation. Sorting can be done using various methods like **Functional vs. Non-functional**, **Priority-based Sorting**, **User Role-based Sorting**, and **Time Dependency Sorting**. Here's how it can be done and where it is used:

1. Functional vs. Non-functional Requirements

- **How**: Separate the requirements into:
    - **Functional Requirements**: Describe what the system should do.
    - **Non-functional Requirements**: Define system attributes like performance, usability, and scalability.
- **Example**:
    - Functional: "The system should allow users to add products to the cart."
    - Non-functional: "The system should process transactions within 3 seconds."
- **Where Used**: This sorting helps during the **design and development phases** to implement critical functionalities first while ensuring performance and quality attributes are met.

2. Priority Sorting

- **How**: Group requirements by priority levels:
    - High Priority: Critical to the system's core functionality.
    - Medium Priority: Important but not immediately required.
    - Low Priority: Nice-to-have features for future releases.
- **Example**:
    - High Priority: "Users must be able to register and log in to the system."
    - Low Priority: "Users can customize their profile pictures."
- **Where Used**: This is used in **agile development** and **release planning**, ensuring that high-priority features are delivered first.

3. User Role-based Sorting

- **How**: Organize requirements by user roles, e.g., admin, customer, vendor, etc.
- **Example**:
    - Admin Role: "Admins can manage user accounts and access reports."
    - Customer Role: "Customers can browse products and place orders."
- **Where Used**: This is useful in the **requirements elicitation and design phase**, ensuring each role's specific needs are addressed.

4. Time Dependency Sorting

- **How**: Categorize requirements based on when they should be implemented:
    - Short-term: Features needed for the initial launch.
    - Long-term: Features for future iterations.
- **Example**:
    - Short-term: "Implement a payment gateway for transactions."
    - Long-term: "Introduce AI-based product recommendations."
- **Where Used**: Helps in **project planning and roadmap creation** to deliver features iteratively.

Where We Use Requirement Sorting

- **Project Planning:** Ensures that critical and time-sensitive requirements are addressed first.
- **Resource Allocation:** Aligns resource distribution based on priority and complexity.
- **Development Cycles:** Guides sprint or iteration planning in Agile methodologies.
- **Stakeholder Communication:** Provides a clear structure to discuss and approve requirements.

Example of Sorting Requirements for an E-commerce Website

1. Functional vs. Non-functional Requirements

- **Functional Requirements**:
    - Users should be able to browse products by categories.
    - The system should allow users to register, log in, and manage accounts.
    - Users should be able to add products to the shopping cart and proceed with the checkout.
    - Admin should be able to manage product listings and inventory.
- **Non-functional Requirements**:
    - The website should load within 2 seconds on average.
    - The system should handle at least 1,000 concurrent users.
    - The website should be compatible with mobile and desktop browsers.

Where Used: Helps the development team prioritize critical functionalities while ensuring the website meets performance and security expectations.

2. Priority-based Sorting

- **High Priority**:
    - User registration and login functionality.
    - Product search and browsing capability.
    - Secure payment gateway integration.

    - Order confirmation with email notification.
- **Medium Priority**:
    - Wishlist or "Save for Later" functionality.
    - Product reviews and ratings.
    - Multiple language support.
- **Low Priority**:
    - Gamification features like loyalty points.
    - Integration with social media for sharing products.

Where Used: Useful for project planning and ensuring critical features are delivered in the first phase, with enhancements in later releases.

3. User Role-based Sorting

- **Customer Role**:
    - Search and browse products.
    - Add products to the cart and complete payment.
    - Track order delivery status.
- **Admin Role**:
    - Add, edit, or delete products.
    - View and manage customer orders.
    - Generate sales reports.

Where Used: Helps design workflows and user interfaces tailored to specific roles

4. Time Dependency Sorting

- **Short-term Requirements (MVP)**:
    - User registration, login, and account management.
    - Product browsing, searching, and cart functionality.
    - Integration of a secure payment gateway.
    - Email notifications for order confirmation.
- **Long-term Requirements**:
    - AI-based product recommendations.
    - Support for multiple currencies.
    - Integration with third-party logistics for real-time delivery tracking.

Where Used: Assists in iterative development and setting project milestones.

Benefits of Sorting Requirements

- Improves focus on critical features.
- Avoids resource waste on low-value tasks.
- Ensures alignment with project goals and stakeholder expectations.

- Supports efficient project delivery and timeline adherence.

Q10. Prioritise the Requirements – –Where we will use? Give one example - 5 Marks

Answer: Prioritizing requirements is a crucial process in business analysis that helps ensure that the most important and high-value requirements are addressed first. In any project, especially when resources (time, budget, and manpower) are limited, not all requirements can be implemented at once. Therefore, prioritization helps to allocate resources effectively and meet key business objectives.

Where We Will Use Prioritizing Requirements?

Prioritizing requirements is used in various stages and contexts throughout the project lifecycle:

1. **Requirement Gathering & Elicitation**:
   - During the early stages of a project, requirements are gathered from different stakeholders. Not all requirements can be delivered in the same release, so they need to be prioritized to align with business goals and user needs.
2. **Agile Development**:
   - In Agile methodologies, the backlog of user stories and features must be prioritized for each sprint or release. The Product Owner or BA ensures that the team works on the most important features first to deliver maximum value in the shortest time.
3. **Project Planning and Roadmap Creation**:
   - Prioritization helps in creating project roadmaps that outline when certain features or capabilities will be delivered based on their importance and dependencies.
4. **Resource Allocation and Time Management**:
   - When resources (time, budget, and personnel) are limited, prioritizing requirements ensures that the most crucial features or functionalities are developed first, especially when trade-offs are necessary.
5. **Stakeholder Management and Communication**:
   - Stakeholders often have differing opinions on what features are most important. By prioritizing requirements, the BA can manage stakeholder expectations and ensure alignment on which features are delivered first.

Methods to Prioritize Requirements

MoSCoW Method of Requirement Prioritization

The **MoSCoW Method** is a popular technique used to prioritize requirements in a project. It categorizes requirements into four groups:

1. **Must Have**: Essential requirements without which the system cannot function or meet the project objectives.
2. **Should Have**: Important requirements, but not critical for immediate delivery; workarounds might exist.
3. **Could Have**: Desirable requirements that add value but are not essential and can be deferred if time or resources are limited.
4. **Won't Have (for now)**: Requirements that are agreed not to be included in the current scope but might be considered in future releases.

Example of Prioritizing Requirements

Example: E-Commerce Website

1. Must Have (Critical Features)

- User registration and login functionality.
- Product browsing and search.
- Shopping cart and checkout process.
- Integration of a secure payment gateway.
- Order confirmation with email notifications.

2. Should Have (Important but Not Critical)

- Product reviews and ratings.
- Support for multiple payment options (e.g., UPI, COD, Credit/Debit cards).
- Basic delivery tracking.

3. Could Have (Nice-to-Have Features)

- Wishlist or "Save for Later" functionality.
- AI-driven product recommendations.
- Multi-language support.
- Integration with social media for product sharing.

4. Won't Have (for now) (Out of Current Scope)

- AR-based product previews.
- Integration with blockchain for payment security.
- Advanced analytics dashboard for customers (e.g., spending habits).

Q11. Weekly status reporting – How we will drive? 5 Marks

Answer: Weekly status reporting is an essential activity in project management, especially for business analysts (BAs) and project teams. It helps track the progress of the project, highlights risks or issues, ensures alignment with business goals, and provides transparency to stakeholders.

Driving a weekly status report effectively ensures that the project remains on track and that everyone is aware of the current status, upcoming tasks, and challenges.

Weekly status reporting is a structured way to track the progress of a project, communicate updates to stakeholders, identify risks, and align teams. Here's how to effectively drive weekly status reporting:

Components of a Weekly Status Report

1. **Title and Reporting Period:** Identify the project name and reporting week.
2. **Executive Summary:** A concise overview of progress, risks, and next steps.
3. **Key Milestones Achieved:** List accomplishments during the week.
4. **Current Status:**
    - Tasks completed.
    - Tasks in progress.
    - Tasks delayed (with reasons).
5. **Risks and Issues:**
    - Current challenges.
    - Potential risks.
    - Mitigation strategies.
6. **Next Steps:**
    - Planned tasks for the upcoming week.
    - Expected outcomes or deliverables.
7. **Action Items and Responsibilities:**
    - Assign owners to each task or action item.

Question that can be asked in weekly status report:

- What milestones or goals were achieved this week?
- What tasks were completed, and are they on schedule?
- Are there any deliverables pending from the previous week?

- What is the overall status of the project (On track, At risk, Delayed)?
- Which tasks or activities are currently in progress?
- Are any tasks ahead of or behind schedule?

- Are there any challenges or obstacles faced this week?
- What risks have been identified, and how are they being managed?
- Are there any unresolved issues requiring escalation or additional support?
- How do these risks or issues affect the timeline, budget, or scope?

- Are there any dependencies delaying progress?
- Is the team waiting on approvals, resources, or information to proceed?
- Are any external factors or stakeholders causing bottlenecks?

- Is the team adequately resourced to meet project objectives?
- Are there any skill gaps or capacity issues impacting progress?
- Are there any team concerns that need to be addressed?

- Is the project within budget?
- Were there any unexpected expenses this week?
- Are there any financial risks to address?

- What are the main priorities and deliverables for the upcoming week?
- Are there any upcoming deadlines or milestones?
- What actions or tasks need stakeholder approval to proceed?

- Are there decisions needed from stakeholders?
- Are there any changes in project scope or requirements?
- What feedback or suggestions do stakeholders have?

- Is the project on track to meet its overall deadline?
- Are there any deliverables that might need to be reprioritized?
- Is there a need to adjust timelines due to delays?

- What went well this
- week that can be repeated?
- What could have been done differently to improve outcomes?

Q12. Meeting Minutes Document – prepare one Sample -5 Marks

Answer: A **Meeting Minutes Document** is an official record of discussions, decisions, and actions agreed upon during a meeting. It serves as a reference for participants and stakeholders and ensures accountability for tasks assigned.

A **Minute of Meeting (MoM)** is prepared **after a meeting is conducted** to document the discussions, decisions, and action points. It serves as a written record of what transpired during the meeting and is typically created and shared shortly after the meeting to ensure accuracy and timely follow-up.

Benefits of a Minute of Meeting:

- Provides a clear summary of the meeting.
- Tracks accountability by documenting tasks and deadlines.
- Acts as a historical record for future reference.

- Helps keep absent members updated.

Structure of a Meeting Minutes Document

1. **Title and Basic Details**
    - **Meeting Title:** Name or purpose of the meeting (e.g., "Project Kickoff Meeting").
    - **Date and Time:** When the meeting occurred.
    - **Location:** Physical or virtual meeting platform (e.g., Zoom, Office Conference Room).
    - **Attendees:** Names and roles of participants.
    - **Meeting Facilitator:** The person leading the meeting.

2. **Agenda**
    - A brief outline of the topics planned for discussion.
    - Example:
        1. Welcome and Introductions
        2. Project Updates
        3. Risk Review
        4. Next Steps

3. **Discussion Points**
    - Detailed notes on the topics discussed.
    - Summarize key points, without unnecessary details.
    - Include any challenges or debates raised during the meeting.

4. **Decisions Made**
    - Record any conclusions or agreements reached.
    - Example: "Approved budget increase for marketing by 10%."

5. **Action Items**
    - List tasks assigned during the meeting.
    - Include:
        - Task Description
        - Assigned Person(s)
        - Deadline
    - Example:
        - Task: "Complete market research report."
        - Assigned To: "John Doe."
        - Deadline: "Jan 20, 2025."

6. **Follow-Ups**
    - Specify any follow-up meetings or actions required.
    - Example: "Schedule a risk review meeting for Jan 15, 2025."

7. **Closing Remarks**
    - Summary of the meeting.
    - Thank participants for their time and contributions.

8. **Prepared By**
    - Name and role of the person who recorded the minutes.
    - Example: "Prepared by Jane Smith, Project Coordinator."

Meeting Title: Sprint Planning Meeting

Date and Time: January 11, 2025, 2:00 PM – 3:00 PM
Location: Microsoft Teams
Attendees:

- Alice Carter (Scrum Master)
- Bob Harris (Product Owner)
- Chloe Adams (Developer)
- David Lee (QA Analyst)

Agenda:

1. Review of Previous Sprint
2. Prioritize Backlog Items
3. Assign Tasks for Next Sprint
4. Identify Dependencies and Risks

Discussion Summary:

- **Previous Sprint Review:** Completed 85% of planned tasks; remaining tasks were deprioritized due to a critical bug fix.
- **Backlog Prioritization:** High-priority tasks identified for the upcoming sprint include API development and front-end integration.
- **Task Assignment:** Team discussed workload distribution and assigned tasks based on individual expertise.
- **Dependencies and Risks:** Highlighted the dependency on external APIs, with a risk of delayed integration.

Decisions Made:

1. Sprint duration will be set at two weeks (Jan 15 – Jan 29, 2025).
2. Prioritize resolving dependencies on external APIs by Jan 20, 2025.

Action Items:

Action Item	Owner	Due Date
Finalize API endpoint documentation	Bob Harris	January 14, 2025
Begin front-end integration work	Chloe Adams	January 15, 2025
Test API endpoints for functionality	David Lee	January 20, 2025
Follow up with external API provider	Alice Carter	January 13, 2025

Agenda Summary:

The meeting focused on finalizing the sprint backlog, addressing dependencies, and assigning tasks to team members for the upcoming sprint.

Next Meeting:

Meeting Title: Sprint Review Meeting
Date and Time: January 29, 2025, 3:00 PM – 4:00 PM
Location: Microsoft Teams
Expected Attendees:

- Alice Carter
- Bob Harris
- Chloe Adams
- David Lee

Q13. Change Tracker – Document - – prepare one Sample -4 Marks

Answer: A **Change Tracker Document** is a structured record used in project management to track, document, and monitor all changes requested or implemented during a project. It helps ensure transparency, proper communication, and accountability among stakeholders and team members regarding changes that impact the project scope, timeline, cost, or deliverables.

Purpose of a Change Tracker Document

1. **Track Changes:** Keep a detailed record of all change requests throughout the project lifecycle.

2. **Analyze Impact:** Assess the effect of changes on project scope, schedule, and budget.
3. **Facilitate Decision-Making:** Help stakeholders approve or reject changes based on their impact and feasibility.
4. **Maintain Accountability:** Record who requested, approved, or implemented a change.
5. **Ensure Documentation:** Keep an auditable history of changes for future reference or compliance purposes.

Date	Version No.	Document Change Description	Name	Title	Signature	Approved By
10-Ja0n-25	v1.1	Added a "Buy Later" feature section to the requirements.	Kevin Miller	Stakeholder	[Signed]	Mr. Henry
12-Jan-25	v1.2	Updated login process to include mobile number option.	Mr. Vandanam	Project Manager	[Signed]	Mr. Karthik
15-Jan-25	v1.3	Included net banking option in the payment gateway.	Ben Taylor	Stakeholder	[Signed]	Mr. Pandu
20-Jan-25	v1.4	Modified delivery tracking system for real-time updates.	Ms. Juhi	Senior Java Developer	[Signed]	Mr. Dooku

Explanation of Columns

1. **Date:** When the change was made or approved.
2. **Version No.:** The version of the document after the change.
3. **Document Change Description:** Summary of what was changed or updated.
4. **Name:** The name of the individual who requested or made the change.
5. **Title:** The role/title of the individual requesting or making the change.
6. **Signature:** Signature of the person confirming the change.
7. **Approved By:** Name of the person who approved the change.

Q14. Difference between Traditional Development Model and Agile Development Models – 8 Marks

Answer: **Traditional Development Model**

The **Traditional Development Model**, such as the Waterfall or V-Model, follows a **sequential and linear process**. Each phase is completed before moving to the next, and once a phase is finalized, it's hard to go back and make changes.

Key Features:

1. **Phased Approach:** Steps include Requirements Gathering, Design, Development, Testing, and Deployment.
2. **Rigid and Structured:** Changes after requirements are defined are difficult to incorporate.
3. **Focus on Documentation:** Detailed documentation is created at each phase, which ensures clarity but can slow down the process.
4. **Late Testing:** Testing happens after the development phase, meaning issues are detected late in the project.
5. **Delivery:** The complete product is delivered at the end of the project.

Best for:
Projects with **well-defined, stable requirements** and where timelines and budgets are strictly fixed, such as government or construction projects.

Agile Development Model

The **Agile Development Model** is a **flexible, iterative, and collaborative approach** where the project is broken into smaller cycles called **sprints** (usually 2-4 weeks). Agile focuses on **incremental delivery** of working software.

Key Features:

1. **Iterative Approach:** The project evolves through multiple iterations, and feedback is incorporated continuously.
2. **Customer Collaboration:** Customers and stakeholders are closely involved throughout the development cycle.
3. **Flexibility:** Agile welcomes changes, even late in the process.
4. **Frequent Delivery:** A small, functional piece of the product is delivered at the end of each sprint.
5. **Focus on Working Software:** Emphasizes delivering functional software over comprehensive documentation.

Best for:
Projects with **dynamic or evolving requirements**, where time-to-market is critical, such as mobile apps or software startups.

Aspect	Traditional Development Model	Agile Development Model

Approach	Follows a sequential process (e.g., Waterfall, V-Model).	Iterative and incremental approach.
Flexibility	Rigid, with predefined phases and little flexibility.	Highly flexible and adaptive to changes.
Delivery	Entire product is delivered at the end of the project.	Working software is delivered in small iterations.
Change Management	Difficult and costly to accommodate changes mid-project.	Changes are welcomed, even late in the process.
Testing	Done after development is complete.	Continuous testing is part of each iteration.
Documentation	Extensive documentation is required at every stage.	Minimal documentation; focuses on working software.
Team Structure	Teams work separate dev and test teams.	Cross-functional teams collaborate closely.
Aspect	Traditional Development Model	Agile Development Model
Customer Involvement	Limited, mainly during initial and final stages.	Continuous involvement throughout the project.
Risk	High due to late-stage testing and feedback.	Low due to frequent feedback and early issue detection.
Best Suited For	Stable and well-defined requirements.	Dynamic and evolving requirements.

Q15. Explain Brainstorming Technique – Where to use? 5 Marks

Answer: Brainstorming is a group problem-solving and idea-generation technique that encourages participants to share creative and diverse ideas freely in a collaborative environment. The goal is to generate a large number of ideas without criticism or judgment.

It encourages open and free-flowing discussion, enabling participants to think creatively and collaboratively without fear of judgment. The primary goal is to maximize the number of ideas and later refine and prioritize them.

Key Features of Brainstorming:

1. **Encourages Creativity**: Promotes out-of-the-box thinking.
2. **No Criticism Rule**: All ideas are accepted without judgment during the session.
3. **Quantity Over Quality**: Focuses on generating a large number of ideas rather than immediately evaluating them.
4. **Collaboration**: Involves group participation to leverage diverse perspectives.

How to Conduct Brainstorming (Summary Points)

1. **Preparation**
    - Define the objective or problem clearly.
    - Select 5–10 participants with diverse perspectives.
    - Set rules: No criticism, encourage creativity, build on ideas.
    - Appoint a facilitator and prepare tools (whiteboards, sticky notes, etc.).
2. **Conduct the Session**
    - Start with a warm-up or creative exercise.
    - Present the objective to ensure focus.
    - Generate ideas using techniques like Mind Mapping, or Brainwriting.
    - Encourage collaboration and build on shared ideas.
3. **Wrap Up**
    - Organize and group similar ideas.
    - Evaluate and prioritize ideas (e.g., through voting).
    - Document all ideas and discussions.
4. **Follow-Up**
    - Refine the prioritized ideas.
    - Share outcomes with participants and stakeholders.
    - Plan actionable steps to implement the best ideas.

Brainstorming can be used in various scenarios, especially where creativity, collaboration, and innovative thinking are required. Below are key areas where brainstorming is most effective:

1. **Problem-Solving:**
    - Identifying solutions to complex or recurring problems..
2. **Requirements Gathering:**
    - Gathering user needs and expectations during project initiation.
3. **Planning and Strategy:**
    - Generating ideas for strategic decisions, business plans, or marketing campaigns.
4. **Product Design and Development:**
    - Brainstorming new features, functionalities, or enhancements.
5. **Decision-Making:**
    - Exploring multiple alternatives before finalizing a decision.
6. **Team Building and Collaboration:**
    - Encouraging team engagement and promoting creative thinking.
7. **Innovation and Creativity:**
    - Generating new ideas for products, services, or business models.
8. **Process Improvement:**
    - Finding ways to optimize workflows, reduce costs, or increase efficiency.
9. **Conflict Resolution:**
    - Resolving disagreements or finding common ground between stakeholders.
10. **Workshops and Training:**
    - Engaging participants during workshops or team meetings to generate insights.

Case study (Q16 – Q20 ◊ 33 Marks)

TTS Company is a multinational Company giving services on Software development in the BFSI Vertical. They have multiple products available. They have Research and Development Wing, which continuously try to improve the Quality of the products and innovation is their USP, this is helping TTS Company to be in Top 10 List. TTS Company came up one initiative to help their Employees with Loans based on their eligibility. To support this cause, they proposed the development of Employees Loan Management System. The Employees Loan Management System will help an organization to manage a loan for its employees online in an efficient way. Employees can request loans, which will be reviewed by the HR and Accounts departments and then loans will be approved or rejected. In case, the loan is rejected, the employee will be informed of the reason for loan rejection. However, in the case of loan approval, Loan approval terms and conditions, the loan repayment schedule will be provided to the employee. If the employee will agree with the loan offer, terms and condition, and repayment schedule, the loan will be granted to the employee and automatic deduction from employee salary will be made.

Q16. What reports Accounts Departments will generate (minimum 5 reports) – 10 Marks

Answer:

Reports Generated by the Accounts Department for the Employees Loan Management System

1. **Loan Approval Report**
    - **Purpose:** A detailed report of all loans approved during a specific period.
    - **Details:**
        - Employee Name and ID
        - Loan Amount Approved
        - Approval Date
        - Loan Tenure
        - Interest Rate
2. **Loan Rejection Report**
    - **Purpose:** A report listing all rejected loan applications along with reasons for rejection.
    - **Details:**
        - Employee Name and ID
        - Loan Amount Requested
        - Rejection Date
        - Reason for Rejection (e.g., ineligibility, insufficient documents)
3. **Loan Repayment Schedule Report**
    - **Purpose:** A report tracking repayment schedules and deductions made from employee salaries.
    - **Details:**
        - Employee Name and ID

        - Monthly Deduction Amount
        - Repayment Start and End Dates
        - Remaining Balance

4. **Outstanding Loans Report**
    - **Purpose:** A report showing all loans that are still being repaid, highlighting overdue repayments if any.
    - **Details:**
        - Employee Name and ID
        - Outstanding Amount
        - Overdue Amount (if applicable)
        - Loan Tenure and Interest Rate
5. **Loan Disbursement Summary Report**
    - **Purpose:** A summary of all loans disbursed within a specific period to track financial outflow.
    - **Details:**
        - Total Loan Amount Disbursed
        - Number of Loans Approved
        - Average Loan Amount Per Employee
        - Department-Wise Loan Distribution
6. **Employee Eligibility Report** (Optional)
    - **Purpose:** A report listing employees eligible for loans based on predefined criteria.
    - **Details:**
        - Employee Name and ID
        - Eligibility Status
        - Loan Limit

These reports will help the Accounts Department ensure accurate financial tracking, maintain compliance, and provide transparency in loan-related activities.

Q17. What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is rejected? – 5 Marks Structure of the Message/Mail for Loan Rejection

Answer:

Subject: Loan Request Rejection Notification

Dear [Employee Name],

We appreciate your application for a loan under the Employees Loan Management System. However, after careful review of your request, we regret to inform you that your loan application has been **rejected** due to the following reason(s):

- [Reason for rejection, e.g., "Ineligibility due to insufficient tenure" or "Loan amount exceeding eligibility criteria."]

Please feel free to reach out to the HR department or Accounts team for further clarification regarding your application.

We encourage you to review your eligibility and reapply in the future if applicable.

Thank you for your understanding.

Best Regards,
[HR Representative's Name]
HR Department
TTS Company
[Contact Details]

Q18. What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is approved? – 5 Marks

Answer:

Subject: Loan Request Approval Notification

Dear [Employee Name],

We are pleased to inform you that your loan application under the Employees Loan Management System has been **approved**. Below are the details of your loan:

Loan Details:

- **Loan Amount Approved:** [Amount]
- **Loan Tenure:** [Duration in months/years]
- **Interest Rate:** [Rate]
- **Monthly Deduction Amount:** [Amount]
- **Repayment Start Date:** [Date]

Next Steps:

1. Please review the **Loan Terms and Conditions** and the **Repayment Schedule** attached to this email.
2. Confirm your acceptance by replying to this email or signing the attached agreement document.

Once we receive your confirmation, the loan will be disbursed, and the repayment will commence as per the agreed schedule.

If you have any questions or need further clarification, please do not hesitate to contact us.

Congratulations on your loan approval!

Best Regards,
[HR Representative's Name]
HR Department
TTS Company
[Contact Details]

Q19. Design a sample report on the Loans applications Received by the accounts department – 8 Marks

Answer: Loan Applications Report

Period: [Start Date] to [End Date]
Prepared By: Accounts Department
Report Date: [Date]

This report provides a summary of all the loan applications received by the Accounts Department within the specified period. It includes loan details, application status, and other relevant information to track the loan approval process efficiently.

Summary of Loan Applications

Application ID	Employee Name	Department	Loan Amount Requested	Application Date	Loan Status	Approval/Rejection Date	Reason for Rejection (if applicable)	Repayment Period	Interest Rate
1	John Doe	IT	₹2,00,000	05-Jan-25	Approved	10-Jan-25	N/A	24 months	8%
2	Jane Smith	HR	₹1,50,000	06-Jan-25	Rejected	12-Jan-25	Insufficient tenure	N/A	N/A
3	Mike Johnson	Finance	₹5,00,000	07-Jan-25	Approved	14-Jan-25	N/A	36 months	7%

4	Emma White	Operations	₹1,00,000	08-Jan-25	Pending	N/A	N/A	N/A	N/A
5	Chris Brown	Sales	₹3,00,000	09-Jan-25	Approved	15-Jan-25	N/A	18 months	9%

Report Summary

- **Total Applications Received:** 5
- **Total Loan Amount Requested:** ₹12,50,000
- **Total Applications Approved:** 3 (Loan Amount: ₹10,00,000)
- **Total Applications Rejected:** 1 (Loan Amount: ₹1,50,000)
- **Total Applications Pending:** 1 (Loan Amount: ₹1,00,000)

Loan Application Status Breakdown

Loan Status	Count of Applications	Total Amount Requested
Approved	3	₹8,00,000
Rejected	1	₹1,50,000
Pending	1	₹1,00,000

Additional Comments:

- **Applications Approved:** The loans for employees John Doe, Mike Johnson, and Chris Brown were successfully approved with a total loan amount of ₹8,00,000.
- **Applications Rejected:** The application from Jane Smith was rejected due to insufficient tenure.
- **Pending Applications:** Emma White’s loan application is still under review by the Accounts Department.

Q20. Which reporting Tools we will use for generating reports. – 5 Marks

Answer: Reporting Tools for Generating Reports

1. **Microsoft Excel**
   - **Description:** A widely used tool for generating reports due to its versatility, ease of use, and robust features like pivot tables, charts, and data filtering.
   - **Use Case:** Ideal for generating simple to complex reports such as loan application summaries, loan approval statuses, and financial reports.
   - **Benefits:**

        - User-friendly interface.
        - Ability to manage large datasets with formulas and functions.
        - Easy integration with other data sources (e.g., databases, external files).
2. **Power BI (Business Intelligence)**
    - **Description:** A Microsoft tool for data visualization and business intelligence, used to create interactive reports and dashboards.
    - **Use Case:** Excellent for generating dynamic, real-time reports with data visualizations such as graphs, charts, and dashboards for loan applications, approval rates, and repayment tracking.
    - **Benefits:**
        - Real-time data refresh.
        - Integration with multiple data sources.
        - Interactive dashboards with drill-down capabilities.
3. **Tableau**
    - **Description:** A powerful data visualization tool that allows for the creation of interactive and shareable dashboards.
    - **Use Case:** Useful for creating visual reports and dashboards for loan approval trends, financial summaries, and loan repayment progress.
    - **Benefits:**
        - Data visualization with drag-and-drop features.
        - Can handle large datasets.
        - Easy sharing of reports across the organization.
4. **Google Data Studio**
    - **Description:** A free data visualization tool that allows users to create customizable and interactive reports.
    - **Use Case:** Best for smaller teams or organizations that need a cost-effective solution for generating dynamic loan application reports and approval/rejection rates.
    - **Benefits:**
        - Cloud-based, accessible anywhere.
        - Integration with Google Sheets and other data sources.
        - Simple to use and share with teams or stakeholders.
5. **SQL Server Reporting Services (SSRS)**

- **Description**: A server-based reporting tool that works with SQL databases.
- **Best For**: Database-driven reporting with static and paginated reports.
- **Key Features**:
    - Ideal for generating detailed reports from structured data.
    - Integration with Microsoft tools and other data warehouses.
    - Export options (PDF, Excel, Word).
- **Use Case**: Generating detailed reports on loan data stored in SQL databases.