**Capstone 3 Part 2**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Brainstorming** | **JAD Session (Joint Application Design)** |
| **1** | **Goal** | Generate a wide range of creative ideas. | Arrive at a well-defined set of requirements and solutions. |
| **2** | **Structure** | Informal and loosely structured. | Structured and focused on specific objectives. |
| **3** | **Participants** | Diverse group of stakeholders (optional). | Project stakeholders and technical teams (required). |
| **4** | **Preparation** | Minimal preparation required. | Pre-defined agenda, documentation, and data analysis. |
| **5** | **Output** | List of raw ideas, often unrefined. | Defined requirements, design decisions, and action items. |
| **6** | **Focus** | Quantity and variety of ideas. | Quality and feasibility of solutions. |
| **7** | **Decision-Making** | Not the primary focus. | Decisions are made collaboratively and documented. |
| **8** | **Example** | Brainstorming new marketing campaign ideas. | JAD session to define requirements for a new software system. |

**Q2. Why Document Analysis is one of the compulsory techniques we use in a Project?**

Document analysis is a crucial technique in project management. It helps extract valuable insights, information, and context, which are essential for project success.

**Understanding Requirements:** Documents provide details about project objectives, scope, and expectations. This helps in aligning all stakeholders and ensuring everyone is on the same page.

**Quality Assurance:** Documents include quality standards, guidelines, and procedures.

These define the expectations for project deliverables, ensuring that outcomes meet predefined standards.

* Understanding the Current State: Documents provide valuable insights into the existing system, business processes, and technical infrastructure. This knowledge is crucial for:
  + Identifying project requirements and ensuring they align with existing systems.
  + Assessing potential risks and challenges associated with changes.
  + Understanding limitations and capabilities of the current system.
* Gathering Requirements: Existing documentation often outlines system functionalities, business rules, data definitions, and user workflows. Analysing these documents helps elicit requirements for the new system or modification, saving time and effort compared to starting from scratch.
* Identifying Gaps and Issues: Reviewing documents can reveal gaps between documented processes and actual practices, highlighting areas for improvement. This can inform the scope of the IT project, ensuring it addresses existing problems.
* Knowledge Transfer: Documents act as a knowledge repository, facilitating knowledge transfer between team members and across project phases. This ensures everyone has a common understanding of the system and project goals.
* Reduced Risk: By analysing documents, you can identify potential risks early on, such as data migration challenges, compatibility issues, or integration complexities. This allows for proactive mitigation strategies.
* Compliance: Many IT projects need to adhere to specific regulations or standards. Document analysis helps ensure the project adheres to these requirements, avoiding legal or compliance issues later on.
* Improved Communication: Documents provide a common reference point for discussions and decision- making. Analysing them fosters a shared understanding among stakeholders and technical teams.
* Reduced Rework: By clearly understanding existing systems and requirements upfront, you can reduce the risk of rework due to mismatched expectations or functionality gaps.

**Q3. - In which context we will use reverse engineering?**

Reverse engineering is used when analysing an **existing system, process, or product** to understand its components, functionality, and structure. It helps when:

* No documentation exists for an old system.
* Upgrading or replacing legacy systems.
* Analysing competitors’ products for benchmarking.
* Understanding business processes in mergers & acquisitions.

It is widely applied across various industries for different purposes, such as:

**Software Development:** To study a program's code, functionality, or design to identify vulnerabilities, create compatible software, or improve performance.

**Product Design and Manufacturing:** To analyse a physical product to understand its design, materials, or assembly process, often for improving the design, creating a similar product, or troubleshooting.

**Cybersecurity:** To examine malware or other harmful software to understand its behaviour, detect vulnerabilities, or develop defensive mechanisms.

**Intellectual Property:** To ensure compliance, check for patent infringement, or recreate older systems or components no longer available.

**Types of Reverse Engineering:**

**Black Box Reverse Engineering**: This involves studying the product's behaviour, input, and output without accessing the internal structure or code. For example, analysing a software's functionality by observing how it reacts to various inputs.

1. **Focus**: Examines external behaviour without accessing the internal workings of the system.
2. **Methods**: Focuses on observing inputs and outputs, running tests, and analysing performance or interaction.
3. **Tools**: May involve monitoring tools, debugging utilities, and recorders to capture system responses.
4. **Use Cases**: Commonly used to:

* Understand functionality without violating access restrictions.
* Test for compatibility and interoperability.
* Analyse competitive software or hardware.
* Detect vulnerabilities through exploratory testing.

1. **Limitations**: Provides no insights into internal code, algorithms, or system design.

*Example:* Testing how a competitor’s app processes and responds to user inputs.

**White Box Reverse Engineering**: This is a deeper analysis where the internal workings, such as the source code, hardware components, or detailed design, are examined. This often requires direct access to the system's internals.

1. **Focus**: Analyzes the internal structure, code, and mechanics of the system.
2. **Methods**: Involves decompilation, disassembly, and code analysis, often requiring authorized access to source code or hardware.
3. **Tools**: Includes disassemblers, decompilers, circuit analyzers, and software debugging tools.
4. **Use Cases**: Typically utilized for:
   * Debugging software or systems.
   * Understanding legacy systems to replicate or enhance them.
   * Malware analysis to identify harmful code and create defenses.
   * Ensuring intellectual property compliance or patent analysis.
5. **Limitations**: Requires advanced tools, expertise, and often permissions to access underlying architecture.

*Example*: Inspecting the source code of an outdated system to recreate or modernize it.

**Q4. What is the difference between Brainstorming and Focus Groups?**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Brainstorming** | **Focus Groups** |
| **Definition** | A technique to generate a large number of creative ideas in a free-flowing discussion. | A structured discussion where participants provide feedback, opinions, and insights on a specific topic. |
| **Goal** | To **generate new ideas** and solutions. | To **gather opinions, preferences, and feedback** from a target audience. |
| **Structure** | Informal and loosely structured. | Moderated and structured with predefined questions. |
| **Participants** | A diverse group, often internal team members. | A selected group representing the target audience (customers, stakeholders, etc.). |
| **Moderation** | Facilitator encourages free thinking without criticism. | Moderator asks specific questions and guides the discussion. |
| **Focus** | Quantity and diversity of ideas. | Quality of insights and in-depth opinions. |
| **Decision-Making** | Not a primary focus; goal is to generate ideas. | Helps in making data-driven decisions. |
| **Example** | Brainstorming new product features within a team. | Conducting a focus group with customers to get feedback on a product prototype. |

**Q5. Observation Technique – Explain both Active and Passive approaches**

Observation techniques play a crucial role in understanding user behaviour and system interactions. Here's a breakdown of two common approaches: Active Observation and Passive Observation:

**Active Observation:**

Goal: Gain a deeper understanding of a user's thought process and decision-making during their interaction with a system.

Process:

* Collaborate with the user beforehand to explain the observation and obtain consent.
* Observe the user as they perform tasks within the system, asking clarifying questions to understand their thought process.
* Encourage the user to "think aloud" and verbalize their thought process as they interact with the system.

Benefits:

* Provides valuable insights into user experience (UX) issues, usability problems, and decision-making patterns.
* Helps identify areas where users might get confused or encounter difficulties.
* Allows for real-time feedback and clarification of observed behaviour.

Example:

An IT professional observes a user trying to complete a complex online form. By asking questions and prompting the user to think aloud, they can understand why the user is hesitating at certain points and identify potential improvements to the form's design or instructions.

**Passive Observation:**

Goal: Observe user behaviour without directly interacting with them. This can be useful for gathering more objective data on user interaction patterns.

**Process:**

* Set up a way to observe user interactions (e.g., screen recording software, eye-tracking tools).
* Observe user behaviour without interfering or prompting them.
* Analyse the captured data to identify patterns and trends in user interaction.

**Benefits:**

* Provides unbiased data on how users actually interact with the system.
* Can be used to measure task completion times, identify frequently accessed features, and uncover common navigation patterns.
* Less disruptive to the user's workflow compared to active observation.

**Example:**

A website developer uses heatmaps to passively observe user interactions on their website. This data reveals which areas of the page users click on the most and which sections they tend to ignore, informing website design improvements.

**Choosing the Right Technique:**

* Use Active Observation when you need to understand user thought processes, decision-making, and identify usability issues.
* Use Passive Observation when you want to gather unbiased data on user interaction patterns and overall user flow within the system.

**Combining Techniques:**

Sometimes, the best approach involves a combination of both active and passive observation. You could use passive observation to gather general user interaction data, and then follow up with active observation sessions with specific user groups to gain deeper insights into their thought processes and decision-making.

By effectively using these observation techniques in IT projects, you can gain valuable insights into user behaviour that can guide the development and improvement of user-friendly and efficient systems.

**Q6. How do you conduct the Requirements Workshop?**

A Requirements Workshop is a structured meeting where stakeholders collaborate to define, clarify, and document business and system requirements. It is widely used in Business Analysis to ensure alignment between business needs and technical implementation.

**Steps to Conduct a Successful Requirements Workshop**

**1. Planning the Workshop**

**Define Objectives** – Identify the scope and goal (e.g., gathering functional requirements for a new software system).  
**Identify Participants** – Include key stakeholders like business users, product owners, developers, and testers.  
**Select a Facilitator** – A Business Analyst or Project Manager typically leads the workshop.  
**Prepare an Agenda** – Structure the session with topics like current challenges, future state, and priorities.  
**Choose Techniques –** Use methods like Brainstorming, JAD (Joint Application Design), Process Flow Diagrams, and Use Case Modelling.

**2. Conducting the Workshop**

**Set the Context** – Explain the purpose and expected outcomes to all participants.  
**Facilitate Discussions** – Ensure everyone actively participates and ideas are documented.  
**Use Visual Aids** – Whiteboards, flowcharts, wireframes, or mind maps help visualize complex processes.  
**Prioritize Requirements** – Use MoSCoW (Must-Have, Should-Have, Could-Have, Won't-Have) or other prioritization techniques.  
**Capture Assumptions & Risks** – Note potential constraints, dependencies, and risks for consideration.

**3. Post-Workshop Activities**

**Document Findings** – Summarize discussions in a Requirements Document, User Stories, or Business Requirement Specification (BRS).  
**Validate with Stakeholders** – Share the documented requirements for review and feedback.  
**Define Next Steps** – Plan follow-up sessions, stakeholder approvals, or refinement meetings if needed.

**Distribute Action Items** - Assign clear action items to participants for any further research, clarification, or approvals.

**Communication** - Share the workshop summaries and refined requirements document with all stakeholders.

**Additional Considerations:**

* Facilitation: Have an experienced facilitator manage the discussion, ensure everyone has a chance to contribute, and keep the session on track.
* Respectful Environment: Encourage open communication, active listening, and respect for diverse viewpoints.
* Time Management: Allocate enough time for each agenda item, but be flexible to adapt to the discussion flow.
* Visual Aids: Use whiteboards, sticky notes, or online tools to promote visual thinking and collaborative problem-solving.
* Manage Expectations: Be upfront about the scope of the workshop and the level of detail you aim to achieve in terms of requirements.

**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.**

A Business Analyst (BA) uses interviews extensively throughout an IT project lifecycle to gather information, understand stakeholder needs, and bridge the gap between business and technical teams.

**Interview Approaches: Structured vs. Unstructured Structured Interviews:**

**Format:** Highly organized with pre-defined questions and answer choices (multiple choice, Likert scale).

**Benefits:**

* + - Ensures standardized data collection across participants, facilitating comparison.
    - Easier to analyse and quantify responses.
    - Saves time as the interviewer follows a set order.
  + **Drawbacks:**
    - Less flexible and can limit in-depth exploration of specific topics.
    - Might not capture nuanced perspectives or unexpected information.

**Unstructured Interviews:**

* + **Format:** Flexible and conversational, allowing the BA to adapt questions based on the interviewee's responses.
  + **Benefits:**
    - Enables in-depth exploration of specific areas of interest.
    - Captures detailed information and personal experiences.
    - Provides a more open environment for sharing ideas and concerns.
  + **Drawbacks:**
    - Time-consuming as the discussion can meander.
    - Can be difficult to compare responses across different interviews.
    - Requires stronger interviewing skills from the BA to guide the conversation effectively.

**Choosing the Right Approach:**

* + **Structured:** Ideal for gathering specific, quantifiable data or when standardized responses are needed from multiple participants.
  + **Unstructured:** Better suited for in-depth exploration of needs, user stories, and gaining qualitative insights into user experiences.

**Open-Ended vs. Closed-Ended Questions Open-Ended Questions:**

* + **Format:** Don't have a predetermined answer. Encourage elaboration and provide rich information.
  + **Examples:**
    - "What are your biggest challenges with the current system?"
    - "How would you like the new system to improve your workflow?"
    - "Can you describe a typical scenario where you encounter difficulty?"

**Closed-Ended Questions:**

* + **Format:** Have a limited set of answer choices or require a yes/no response.
  + **Examples:**
    - "Do you find the current system easy to use?" (Yes/No)

"On a scale of 1 to 5, how satisfied are you with the data reporting capabilities?" (1-5 Likert Scale)

**Question 8- Questionnaire Technique- Where we will use? Give one example.**

Questionnaires are valuable tools in IT projects for gathering information from a wider audience in a cost-effective and efficient way. The questionnaire technique is a method of data collection commonly used in research, surveys, and assessments. It involves presenting a set of written questions to respondents and collecting their responses.

* Questionnaires can be administered in various ways, including:
* Paper and pencil
* Online surveys

**Types of Requirements:**

Face-to-face interviews (where questions are read to participants).

* + **Large-Scale Needs Gathering:** When you need to gather input from a broad population of users, questionnaires are more efficient than individual interviews. This might be relevant for gauging general user satisfaction with an existing system or understanding feature preferences for a new system.
  + **Gathering Specific Data:** Questionnaires can be used to collect specific data points that can be easily quantified and analysed statistically. This could include demographic information, usage patterns of a specific system feature, or user satisfaction ratings on a standardized scale.
  + **Following Up on Interviews:** After initial interviews with key stakeholders, questionnaires can be used to reach a wider audience and gather additional information on specific topics raised during the interviews.
  + **Pre- and Post-Training Assessments:** Questionnaires can assess user knowledge or skills before and after training on a new system, providing data on the effectiveness of the training program.

**Example: User Satisfaction Survey**

**Scenario:** You're a BA working on a project to upgrade a company's customer relationship management (CRM) system. To understand user sentiment towards the current system, you decide to conduct a questionnaire-based survey.

**Questionnaire Content:**

* + **Demographics:** Basic information like department, years of experience using the CRM.
  + **Usage Patterns:** How often do you use the CRM? What features do you use most frequently?
  + **Satisfaction Ratings:** Rate your satisfaction with specific aspects of the CRM (e.g., data entry, reporting, usability).
  + **Open-Ended Questions:** What are the biggest challenges you face with the current CRM? What features would be most helpful in the new system?

**Benefits:**

* + **Cost-Effective:** Reaching a large user base is cheaper than individual interviews.
  + **Standardized Data:** Easier to analyse and compare responses across all users.
  + **Anonymity:** May encourage more honest feedback compared to in-person interviews.

**Question 9. How to Sort the Requirements – Where we will use? Give one example**

Sorting helps structure requirements before prioritization, ensuring a clear understanding for stakeholders and development teams. This makes the prioritization and implementation process smoother. Sorting requirements is essential in requirements management to help with:

* Organization
* Prioritization
* Categorization for better analysis and implementation

Sorting is usually performed during requirements elicitation and documentation phases of a project.

**Ways to Sort Requirements:**

* Functional vs. Non-Functional Requirements – Grouping based on system functionalities vs. performance/security constraints.
* Priority Sorting – Sorting by urgency or business impact.
* User Role Sorting – Categorizing based on different stakeholders (e.g., admin, customer, vendor).
* Time Dependency Sorting – Organizing based on deadlines or phases in a project timeline.

Example: Sorting Requirements for an E-Commerce Website (Sorting Based on Functional vs. Non-Functional Requirements)

|  |  |
| --- | --- |
| **Type** | **Requirement** |
| **Functional** | Users should be able to add products to the cart. |
| **Functional** | Admins should be able to manage inventory. |
| **Non-Functional** | The website should load within 3 seconds. |
| **Non-Functional** | The system should support 500,000 concurrent users. |

**Question 10. Prioritise the Requirements – –Where we will use? Give one example**

Prioritizing requirements is crucial in IT projects to prioritize development efforts, manage project scope, and ensure the most critical functionalities are delivered first. Factors influencing prioritization are Importance, risk, cost, benefit, time and strategy.

**Why Prioritize?**

* + **Focus on Value:** Prioritized requirements ensure that features with the highest business value are developed first, maximizing the return on investment (ROI) of the project.
  + **Manage Dependencies:** Sorting helps identify requirements that depend on others, allowing for a logical development sequence.
  + **Reduce Project Risk:** Prioritization helps manage project scope by identifying features that can be deferred or removed if necessary.
  + **Improved Communication:** Everyone involved (stakeholders, developers, BAs) understands the relative importance of each requirement.

**Where is it used?**

Requirement prioritization is used in **projects and product development** to determine **which features or tasks should be completed first** based on their **value, urgency, and feasibility**.

**Software Development** – Prioritizing features in Agile sprints.  
**Business Process Improvement** – Addressing high-impact changes first.  
**Project Management** – Allocating resources effectively.  
**Compliance & Risk Management** – Ensuring critical security requirements are implemented first.

Example: Prioritizing Requirements for an E-Commerce Website

**Scenario:**

A **Business Analyst** gathers several requirements for an e-commerce website and needs to **prioritize them** based on business value and effort.

**Applying MoSCoW Prioritization**

|  |  |  |
| --- | --- | --- |
| **Priority** | **Requirement** | **Reason** |
| **Must-Have** | Secure payment processing | Essential for transactions. |
| **Must-Have** | User login and registration | Required for customer accounts. |
| **Should-Have** | Order tracking feature | Improves user experience. |
| **Could-Have** | AI-based product recommendations | Nice-to-have but not critical. |
| **Won’t-Have** | Augmented Reality preview | High effort, low immediate value |

**Question 11. Weekly status reporting – How we will drive?**

A weekly status report is a summary of all work completed during the week and how those activities contributed to a task, project, or team objectives. It helps track progress, identify blockers, and align the team on key priorities. Weekly status reporting:

* Improves communication & transparency.
* Helps teams stay aligned on goals.
* Identifies risks early for proactive resolution.
* Keeps leadership informed of project progress.
* **Key Questions to Ask in Weekly Status Reports**

✔ What have you been working on recently? → (Tasks in progress)  
✔ What have you accomplished this week? → (Completed tasks/deliverables)  
✔ What are your top priorities? → (Upcoming focus areas)  
✔ What are your challenges going into next week? → (Roadblocks/risks)

Below are the steps to drive Weekly Status Report effectively

1. Define reporting requirements
2. Set reporting frequency and deadline
3. Standardize reporting format
4. Communicate expectations
5. Provide guidance and support
6. Remind and follow-up
7. Review and consolidate reports
8. Share and discuss reports
9. Act on the findings

**Question 12. Meeting Minutes Document – prepare one Sample**

A **Minutes of Meeting (MoM)** document is a formal written record that captures key discussions, decisions, and action items during a meeting. It ensures that all participants are aligned on the meeting’s outcomes and responsibilities.

**Purpose of MoM:**

**Records Discussions:** Provides a summary of what was discussed in the meeting.  
**Documents Decisions:** Keeps track of important decisions made during the meeting.  
**Assigns Responsibilities:** Clearly defines action items, owners, and deadlines.  
**Ensures Accountability:** Helps participants follow through on commitments.  
**Acts as a Reference:** Serves as an official record for future reference.

|  |  |
| --- | --- |
| Meeting Title | |
| Date and Time |  |
| Location |  |
| Attendees |  |
| Agenda |  |
| Discussion Summary |  |
| Decisions Made |  |
| Action Items |  |
| Owner |  |
| Due Date |  |
| Agenda Summary |  |
| New Meeting | |
| Meeting Title |  |
| Date and Time |  |
| Location |  |
| Expected Attendees |  |

**Question 13. Change Tracker – Document - – prepare one Sample**

Change tracker document is used by the project team to log and track change requests made throughout the life of the project. It helps ensure transparency, accountability, and proper documentation of any modifications made over time.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Change ID** | **Change Requestor** | **Change Description** | **Reason for Change** | **Impact Analysis** | **Approval Status** | **Implementation Date** | **Owner/Responsible Person** | **Change Status** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

OR more detailed:

|  |  |
| --- | --- |
| **Change Tracker Document** | |
| ***Version*** |  |
| ***Date*** |  |
| **Change Details** | |
| ***Change Request***  ***Number*** |  |
| ***Requested By*** |  |
| ***Date of Request*** |  |
| ***Change Description*** |  |
| **Change Assessment** | |
| ***Impact Analysis*** |  |
| ***Risk Analysis*** |  |
| ***Feasibility Analysis*** |  |
| ***Effort Estimate*** |  |
| ***Approval Status*** |  |
| ***Approval Date*** |  |
| **Implementation Details** | |
| ***Developer / Implementer*** | |
| ***Start Date*** | |
| ***End Date*** | |
| ***Test Coverage*** | |
| ***Test Results*** | |
| ***Deployment Plan*** | |
| **Rollback Plan** | |
| ***Rollback Procedure*** | |
| ***Rollback Test Plan*** | |
| ***Rollback Date*** | |
| ***Rollback Results*** | |
| **Documentation Update** | |
| ***Document Updated*** | |
| ***Update Description*** | |
| ***Update Date*** | |
| ***Updated By*** | |
| **Approvals** | |
| ***Approver 1*** | |
| ***Approver 2*** | |
| ***Approver 3*** | |
| ***Approval Date*** | |
| **References** | |
| ***Related Documents*** | |
| ***Supporting Material*** | |

**Question 14. Difference between Traditional Development Model and Agile Development Models**

**Waterfall Model:** The Waterfall model is a linear and sequential approach to software development. It follows a step-by-step process where each phase must be completed before moving to the next. It is used:

* When requirements are well-defined and unlikely to change
* For small or simple projects
* When strict documentation and compliance are required (e.g., banking, healthcare)

**Agile Model:** The Agile model is an iterative and incremental approach where development happens in small cycles (sprints). Agile focuses on continuous feedback and flexibility, allowing changes at any stage of development. Its used:

* When requirements are unclear or changing frequently
* For complex and dynamic projects
* When continuous customer feedback is needed

**Key Differences:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Waterfall Model** | **Agile Model** |
| **Approach** | Sequential | Iterative |
| **Flexibility** | Rigid, hard to change | Flexible, welcomes change |
| **Client Involvement** | Low | High |
| **Delivery** | One-time final product | Continuous, incremental deliveries |
| **Testing** | Done after development | Continuous testing |

**Question 15. Explain Brainstorming Technique – Where to use?**

Brainstorming is a creative problem-solving technique used in IT projects to generate a wide range of ideas in a short period. It encourages participants to think outside the box and come up with innovative solutions.

**Where Brainstorming is useful in IT projects:**

* **Project Kick-off:** Brainstorming is a great way to kick-off a project by stimulating creative thinking and generating a broad spectrum of ideas for functionalities, features, and approaches.
* **Requirements Gathering:** During the initial stages, brainstorming with stakeholders can help identify user needs, pain points, and potential solutions.
* **Problem-Solving:** When encountering roadblocks or technical challenges, a brainstorming session can help the development team explore diverse solutions and approaches.
* **Innovation and Improvement:** Brainstorming can be used to explore new ideas for product features, system enhancements, or process improvements.

**Key Principles of Effective Brainstorming:**

* **Quantity over Quality:** Focus on generating as many ideas as possible, regardless of their initial feasibility.
* **Open Communication:** Encourage open discussion and build upon each other's ideas.
* **Free Flow of Ideas:** No idea is considered bad or off-limits during brainstorming.
* **Encourage Participation:** Create a safe and inclusive environment where everyone feels comfortable sharing their thoughts.
* **Visual Aids:** Use whiteboards, sticky notes, or online tools to capture and visualize ideas collaboratively.

**Example: Brainstorming New Features for a Mobile App**

Imagine you're working on a project to develop a new mobile app for a food delivery service. Here's how brainstorming could be used:

* **Prompt:** "What features would make our food delivery app the most user-friendly and efficient?"
* **Possible Ideas:** Real-time order tracking on a map, personalized restaurant recommendations, in-app chat with customer support, loyalty program integration, integration with popular food review platforms, pre- order scheduling, etc.

**Question 16. What reports Accounts Departments will generate (minimum 5 reports)**

* **Financial Statements:** The accounts department prepares and provides financial statements, including balance sheets, income statements, and cash flow statements. These statements give an overview of the borrower's financial position, profitability, and ability to generate cash flow.
* **Company reserve loan Report**: This report will help understand the reserve amount.
* **Credit Report:** The accounts department may obtain a credit report on the borrower from a credit bureau. This report provides information on the borrower's credit history, including their repayment track record, outstanding loans, and credit score.
* **Collateral Evaluation:** If the loan requires collateral, the accounts department may be involved in evaluating the value and marketability of the proposed collateral. Cash Flow Projections: The accounts department prepares cash flow projections based on the borrower's financial data.
* **Debt-to-Income Ratio Analysis:** The accounts department calculates the borrower's debt-to-income ratio, which compares the borrower's total debt obligations to their income.

**Question 17. What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is rejected?**

Subject: Loan Application Status – Rejected

Dear [Employee Name],

We regret to inform you that your loan application (Reference No: [Loan ID]) submitted on [Date] has been rejected due to the following reason(s):

[Reason for rejection – e.g., insufficient eligibility, pending previous loan, low credit score, etc.]

If you require further clarification, please contact the HR or Accounts department.

Thank you for your understanding.

Best Regards,

[HR Representative Name]

HR Department

TTS Company

**Question 18. What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is approved?**

Subject: Loan Application Status – Approved

Dear [Employee Name],

We are pleased to inform you that your loan application (Reference No: [Loan ID]) submitted on [Date] has been approved.

Loan Details:

Loan Amount: [Amount]

Interest Rate: [Rate]%

Repayment Tenure: [Months]

EMI Deduction: [Amount] per month from salary

Next Steps:

Please review the loan agreement, including the terms and conditions and repayment schedule, attached to this email.

To proceed, kindly sign and return the attached agreement by [Deadline Date]. The loan amount will be disbursed after receiving your confirmation.

For any queries, please reach out to HR or Accounts.

Best Regards,

[HR Representative Name]

HR Department

TTS Company

**Question 19. Design a sample report on the Loans applications Received by the accounts department**

**TTS** Loan Applications Received Report – Accounts Department Date: 22-Mar-25

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Loan ID** | **Employee ID** | **Employee Name** | **Loan Amount** | **Request Date** | **Status** | **Approval Date** | **Rejection Reason  (if any)** |
| L001 | E1023 | Mansi Karan | INR 50000000 | 01-Mar-25 | Approved | 05-Mar-25 | N/A |
| L002 | E1045 | Vedika Saxena | INR 25000000 | 03-Mar-25 | Rejected | N/A | Low Credit Score |
| L003 | E1070 | Alpika Juhi | INR 45000000 | 04-Mar-25 | Approved | 07-Mar-25 | N/A |
| L004 | E1092 | Isha Gupta | INR 20000000 | 06-Mar-25 | Approved | 10-Mar-25 | N/A |
| L005 | E1108 | Aditya Soni | INR 50000000 | 08-Mar-25 | Rejected | N/A | Exceeded Loan Limit |

**Notes:**

* Approved applications have met the loan approval criteria and are eligible for loan disbursement.
* Rejected applications do not meet the loan approval criteria and have been declined.
* Pending applications are currently under review and a decision will be communicated soon.
* For any inquiries or further information, please contact the Accounts Department.

**Question 20. Which reporting Tools we will use for generating reports?**

1. Microsoft Power BI – For interactive dashboards and financial impact analysis.
2. Tableau – For visual representation of loan data and employee credit trends.
3. Excel– A widely used spreadsheet program for collecting, sorting, calculating, tracking and sharing data, including loan-related financial information.