COEPD – Prep Exam 3 –Part 2/2

**Question 1**

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

Brainstorming and Joint Application Development (JAD) sessions are both collaborative techniques used to generate ideas or gather requirements, but they differ significantly in purpose, structure, and application. Here are the key differences:

**JAD :-**

A JAD is a Joint Application Development (or Design) session. It is an opportunity for stakeholders with different points of view to come together to understand business requirements and brainstorm what the best technical approach might be for meeting the customer's needs.

A JAD is a challenging environment because it is a meeting or workshop that brings together people from a number of different disciplines. They could include business or product owners, systems analysts, enterprise architects, solution architects, software developers, and managerial staff. Each will come with their own points of view and may at times resort to lingo specific to their focus area.

The Business Analyst (BA) frequently stands front and center as the facilitator of a [JAD](https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/152/What-is-Joint-Application-Development-JAD.aspx) and as the presenter of what the business needs, objectives, and requirements are. The BA may also further elicit requirements as needed.

**Brainstorming :-**

Brainstorming is a technique that business analysts use to generate ideas for solving problems or opportunities. It's a group activity that encourages free thinking and collaboration

Brainstorming techniques

**Brainwriting**: Have participants write down ideas anonymously on post-it notes or index cards. Then share the ideas randomly and get feedback from the group**Reverse brainstorming**: Consider how to make the problem worse, then reverse those ideas to find solutions

**Mind mapping**: Use visual tools like diagrams to generate and organize ideas

**Purpose**

* **Brainstorming:**
  + Focuses on generating a broad range of ideas or solutions.
  + Typically used for creative thinking and problem-solving.
  + Encourages free-flowing ideas without immediate evaluation.
* **JAD Sessions:**
  + Focus on gathering detailed requirements for a project or system.
  + Aim to bring stakeholders together to ensure consensus and clarity.
  + Structured to create actionable deliverables, such as system specifications.

**2. Participants**

* **Brainstorming:**
  + Usually involves a small group of individuals, often from a single team or department.
  + Participants can be subject matter experts or people with diverse perspectives.
* **JAD Sessions:**
  + Involve key stakeholders, including business users, IT professionals, project managers, and sometimes external consultants.
  + Emphasis on having representatives from all relevant groups for comprehensive input.

**3. Structure**

* **Brainstorming:**
  + Informal and less structured.
  + Participants share ideas openly, often in a freeform manner.
  + No immediate critique or analysis of ideas; focus is on quantity over quality.
* **JAD Sessions:**
  + Highly structured with predefined agendas and facilitators.
  + Include techniques like process mapping, prototyping, and consensus building.
  + Often span multiple sessions with clear objectives for each meeting.

**4. Tools and Techniques**

* **Brainstorming:**
  + Relies on tools like whiteboards, sticky notes, or mind maps.
  + Techniques include round-robin brainstorming, mind mapping, and group discussions.
* **JAD Sessions:**
  + Use tools like flowcharts, use-case diagrams, and modeling tools.
  + Techniques include facilitated workshops, prototyping, and consensus-building activities.

**5. Outcomes**

* **Brainstorming:**
  + Produces a list of ideas or potential solutions.
  + The output often requires further refinement and prioritization.
* **JAD Sessions:**
  + Produces detailed documentation, such as functional requirements, process workflows, and system specifications.
  + The output is directly actionable and feeds into project planning and design.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Brainstorming** | **JAD Sessions** |
| **Purpose** | Generate ideas | Gather detailed requirements |
| **Participants** | Small, diverse group | Stakeholders and representatives |
| **Structure** | Informal, free-flowing | Structured, agenda-driven |
| **Tools/Techniques** | Whiteboards, sticky notes | Flowcharts, use-case diagrams |
| **Outcomes** | List of ideas | Actionable requirements |

**Question 2 :-Why Document Analysis is one of the compulsory technique we use in a Project**

### Document Analysis is a crucial technique in project management and system development for several reasons. It serves as a foundational step in understanding the current state, requirements, and constraints of a project. Here's why it is often considered compulsory:

### 1. Provides Historical Context

* **Understanding the Current State:**
  + Analyzing existing documents, such as policies, procedures, and workflows, helps you understand the current systems, processes, or business environment.
  + This context is critical for identifying areas for improvement or gaps in the current system.
* **Learning from Past Projects:**
  + Documents from previous projects (e.g., lessons learned, reports) provide insights into what worked, what didn’t, and why.
  + Helps avoid repeating past mistakes.

**2. Identifies Requirements**

* **Implicit and Explicit Requirements:**
  + Many project requirements are already documented in user manuals, system specifications, or business process documents.
  + Document analysis ensures these requirements are not overlooked.
* **Regulatory and Compliance Needs:**
  + Analyzing legal and regulatory documents ensures the project complies with applicable laws and standards.
  + Helps avoid legal risks or non-compliance issues.

**3. Saves Time and Effort**

* **Reduces Redundancy:**
  + By analyzing existing documentation, you can avoid duplicating efforts to gather information that already exists.
  + Focuses resources on areas that genuinely need exploration.
* **Speeds Up Requirement Gathering:**
  + Provides a quick way to gather baseline information before engaging stakeholders.
  + Helps prepare for interviews, workshops, or other elicitation techniques.

**4. Validates Stakeholder Inputs**

* **Cross-Referencing Information:**
  + Stakeholder-provided information can be cross-checked against documented policies, standards, or procedures for accuracy.
  + Ensures the requirements are grounded in factual, consistent information.
* **Clarifies Ambiguities:**
  + Documents help clarify or resolve conflicting inputs from stakeholders.

**5. Supports Decision-Making**

* **Informs Design and Planning:**
  + Insights from documents help define project scope, goals, and constraints.
  + Informs decisions about timelines, resources, and project deliverables.
* **Risk Identification:**
  + Identifies risks and challenges based on historical data, operational reports, and audit findings.

**6. Ensures Completeness**

* **Holistic Understanding:**
  + Document analysis ensures no critical details are missed, such as integration points, technical specifications, or stakeholder expectations.
* **Requirement Traceability:**
  + Helps establish a traceability matrix by connecting requirements to source documents.

**Examples of Documents Analyzed in Projects**

* Business Process Diagrams
* Policies and Procedures
* Contracts and Agreements
* User Manuals and System Documentation
* Regulatory and Compliance Guidelines
* Audit Reports and Historical Project Reports

### ****Conclusion****

Document analysis is compulsory because it provides a structured and efficient way to gather, validate, and organize information that is critical for project success. It ensures that the project team starts with a solid understanding of the baseline, reduces risks, and improves the quality of deliverables. Without document analysis, important details could be overlooked, leading to gaps in requirements, scope creep, or project failure.

**Question3 :-**

**In Which Context we will use Reverse Engineering**

Reverse engineering is used in contexts where there is a need to analyze, understand, or recreate an existing system or product. Its applications span industries such as software, hardware, cybersecurity, manufacturing, education, and forensics. However, it must be used responsibly and ethically, adhering to legal and intellectual property regulations.

**Reverse engineering** is used in various contexts, typically when there is a need to analyze or reconstruct an existing system, product, or process. Below are the common contexts where reverse engineering is applicable:

**1. Software Development and Maintenance**

* **Legacy Systems Understanding:**
  + When the original documentation of a system is unavailable, incomplete, or outdated.
  + To understand the architecture, components, and behavior of a legacy application for updates or migrations.
* **Debugging and Bug Fixing:**
  + To analyze software and locate the source of bugs or vulnerabilities.
  + Useful for understanding third-party libraries or software dependencies.
* **Code Optimization and Refactoring:**
  + To extract the underlying logic of poorly written or unoptimized code for improvement.

**2. Product Design and Development**

* **Competitor Analysis:**
  + To study competitors' products for benchmarking or creating improved designs.
  + Ensures innovation while maintaining differentiation.
* **Redesigning or Enhancing Existing Products:**
  + To rebuild a product with improved functionality, efficiency, or compatibility.
  + For creating interoperable systems or components.

**3. Cybersecurity**

* **Vulnerability Analysis:**
  + To analyze malicious software (e.g., viruses, malware) and understand how it operates.
  + Helps in creating effective defense mechanisms and patches.
* **Security Audits:**
  + To assess the robustness of software or hardware against cyber threats.
  + Identifies backdoors or unintended behaviors.

**4. Hardware Analysis**

* **Reproducing Components:**
  + To replicate discontinued or rare hardware parts when the original blueprints are not available.
  + Useful in industries like aviation, manufacturing, and automotive.
* **Integration with Modern Systems:**
  + To analyze older hardware systems and adapt them for compatibility with modern technologies.

**5. Regulatory and Compliance Contexts**

* **Intellectual Property (IP) Verification:**
  + To ensure a product does not infringe on existing patents or copyrights.
  + To verify claims of originality in products or software.
* **Regulatory Compliance:**
  + To ensure that products meet regulatory standards (e.g., safety, emissions, and technical standards).

**. Education and Research**

* **Learning and Training:**
  + To teach students and professionals how systems work by disassembling and analyzing them.
  + Often used in software engineering, hardware design, and cybersecurity courses.
* **Prototyping and Innovation:**
  + To experiment with existing systems to discover new functionalities or applications.

**Question 4**

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

Brainstorming and focus groups are both collaborative techniques used to generate ideas and gather insights, but they differ significantly in purpose, structure, and outcomes. Here's a detailed comparison

While both brainstorming and focus groups are collaborative, brainstorming emphasizes idea generation and creativity within teams, whereas focus groups focus on gathering feedback and insights from external participants to inform decisions. The choice between them depends on the project’s goals and the type of input needed.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Brainstorm** | **Focus Group** |
| Purpose | Generate ideas | Improve existing ideas |
| Trigger | A need to solve a problem | A need to study an existing idea, solution or process |
| Condition | Problem exist | Idea, solution or process exist |
| Number of participants | 8-Jun | 12-Jun |
| Participant types | Heterogeneous | Can be homogeneous or heterogeneous |
| Person running the show | Facilitator | Skilled moderator |
| Knowledge of topic of discussion | Not necessary | In depth knowledge of topic of discussion |
| Guide | Develop criteria for evaluating and rating ideas | Create a discussion guide and moderator scripts |
| Ground rules | Must have | Nice to have |
| Duration | Restrict time to produce ideas. 1-2 Hours | 1 – 2 hrs and sometimes over several days |
| Type of questions to ask | Progressive closed-ended to generate and build on ideas | Can be open-ended  to generate qualitative data or closed-ended to generate quantitative data |
| Observers | No | Yes |
| Result | List of ideas combined to form themes | Report of findings, Could be , bullet list of information learned, Controlled, opinion-focused |
|
|

**Question 5 :-**

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

### ****1. Active Observation****

Active observation involves the observer directly interacting with the participants, engaging in their activities, or asking questions during the observation process.

#### **Key Features:**

* **Engagement:** The observer participates or intervenes in the process to gather insights.
* **Interaction:** The observer may ask clarifying questions, provide input, or even perform some tasks.
* **Purpose:** Often used to gain in-depth understanding and clarify observed behaviors.
* **Examples:**
  + A business analyst sitting with a user to observe and ask about their workflow while performing tasks.
  + A researcher conducting contextual inquiry by actively asking questions during the activity.

#### **Advantages:**

* Provides detailed and nuanced insights by directly engaging with participants.
* Helps clarify ambiguities and understand the reasons behind certain actions or decisions.
* Encourages immediate feedback and discussion.

#### **Challenges:**

* Participants may feel uncomfortable or act differently due to the observer’s involvement (Hawthorne effect).
* The observer’s presence and interaction can inadvertently influence the process or behavior.

### ****2. Passive Observation****

Passive observation involves the observer watching and recording activities without interfering or interacting with participants.

#### **Key Features:**

* **Non-intrusive:** The observer remains unobtrusive and does not engage with participants.
* **Focus on Natural Behavior:** Observes participants in their natural environment without influencing their actions.
* **Purpose:** Often used to gather unbiased, authentic data.
* **Examples:**
  + Observing employees working on a production line without interacting with them.
  + Watching how customers navigate through a store to understand their purchasing behavior.

#### **Advantages:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* Captures genuine and unaltered behavior, as participants are not influenced by the observer’s presence.
* Suitable for sensitive scenarios where interference may disrupt natural workflows.
* Provides insights into unconscious behaviors or patterns.

#### **Challenges:**

* Limited ability to understand the reasons or motivations behind observed actions.
* Observer may miss important contextual information that could have been obtained through interaction.

Comparison:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Active Observation** | **Passive Observation** |
| **Observer Role** | Engages with participants | Watches without interference |
| **Interaction** | High | None |
| **Influence on Process** | Moderate to high | Minimal |
| **Data Collected** | Detailed and explanatory | Unbiased and authentic |
| **Purpose** | To understand motivations and context | To observe natural behavior |
| **Best Use Cases** | When context or reasoning is critical | When unbiased, natural data is required |

### ****When to Use Active vs. Passive Observation****

* **Active Observation:**
  + To explore reasons behind behaviors or decisions.
  + When the observer needs to clarify, learn, or contribute.
  + Suitable for collaborative environments or when building prototypes.
* **Passive Observation:**
  + To gather unbiased data in natural settings.
  + When observing sensitive or high-stakes environments where interference may disrupt workflows.
  + Useful for identifying patterns or behaviors without participant awareness.

### ****Conclusion****

Both active and passive observation have their unique strengths and limitations. Active observation provides deeper insights through engagement, while passive observation captures authentic behavior. The choice between the two depends on the goals of the observation, the environment, and the need for interaction. Often, combining both approaches provides a more comprehensive understanding.

**Question 6 :- How do you conduct the Requirements Workshop ?**

This is a structured meeting with the specifc goal of capturing requirements. It is used to define, prioritize and hopefully Finalize requirements for the new initiative that you’re working on. Requirements workshops typically last between one and a few days. They should also be a highly focused event that is let by a seasoned facilitator.

**Preparation:**

* **Set clear goals and objectives:** Determine what you want to achieve with the workshop, like identifying key features, understanding user needs, or prioritizing requirements.
* **Identify stakeholders:** Invite relevant individuals from different departments, including users, developers, product managers, and business analysts.
* **Develop a detailed agenda:** Outline the topics to be covered, including time allocations for each activity.
* **Choose appropriate facilitation techniques:** Select methods like brainstorming, user story mapping, or affinity diagrams to gather information effectively.
* **Prepare materials:** Create visual aids, sticky notes, whiteboards, and any necessary pre-work documents.

**During the workshop:**

* + **Welcome and introductions:** Set the tone by explaining the workshop purpose and expectations, and allow participants to introduce themselves.
  + **Review project context:** Provide an overview of the project goals, background information, and existing systems.
  + **Elicit requirements:**
    - **Brainstorming:** Generate a comprehensive list of features and functionalities through open discussion.
    - **User stories:** Capture requirements from the user perspective using the "As a [user], I want to [action], so that [benefit]" format
    - **Prioritization:** Use techniques like MoSCoW method to categorize requirements as "Must-Have," "Should Have," "Could Have," and "Won't Have".
  + **Document requirements:** Assign a scribe to capture all discussed features, user stories, and priorities in a clear and organized manner.
  + **Clarify and validate:** Ask clarifying questions to ensure everyone understands the requirements and address any concerns.

**Post-workshop follow-up:**

* + **Summarize and distribute results:** Create a consolidated document outlining all gathered requirements, including priorities and action items.
  + **Get feedback and approval:** Share the summary with stakeholders for review and confirmation.
  + **Track progress:** Monitor the implementation of requirements and address any emerging issues.

### ****Conclusion****

Requirement workshops are a powerful tool for gathering and refining requirements efficiently and collaboratively. When conducted effectively, they promote alignment, stakeholder engagement, and a shared understanding of project goals, significantly increasing the chances of project success.

**Question 7 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 .**

**Context for Conducting Interviews by a Business Analyst (BA)**

An interview is a **valuable technique** for a Business Analyst (BA) to gather in-depth information from stakeholders, users, or subject matter experts (SMEs). Interviews are especially useful when:

* **Gathering Requirements**: To understand business needs, processes, and user expectations.
* **Validating Information**: To confirm details or clarify ambiguities from other data sources.
* **Understanding Challenges**: To identify pain points, opportunities for improvement, or current system issues.
* **Exploring Solutions**: To get feedback on potential solutions or features.
* **Stakeholder Engagement**: To build rapport and trust with key stakeholders.

Interviews allow BAs to collect qualitative data that can be rich and nuanced, which is critical in the early stages of a project or when exploring new opportunities.

### ****Approaches to Conducting Interviews****

There are generally **two types of interview approaches** that can be used in requirements gathering and other BA activities: **Structured** and **Unstructured** interviews.

* **Structured interviews** are helpful when you need consistent data from multiple participants or when gathering factual information, whereas **unstructured interviews** are more beneficial for exploratory discussions and discovering deeper insights.
* **Open-ended questions** provide depth and context, making them great for exploring topics in detail, while **closed-ended questions** allow you to collect specific, measurable information quickly and efficiently.

#### **Structured Interviews**

**Characteristics:**

* **Predefined Questions**: The BA prepares a list of questions ahead of time.
* **Standardized Process**: Every interviewee is asked the same set of questions in the same order.
* **Objective**: Primarily used when you want to gather consistent and comparable data across a large number of participants.
* **Focus**: Narrow, specific, and focused on particular aspects of the project or system.

#### **2. Unstructured Interviews**

**Characteristics:**

* **Flexible and Open-ended**: The BA has a general idea of the topics to cover but allows the conversation to evolve naturally.
* **Flow**: Questions are not rigidly defined, and the direction of the interview may change based on the interviewee's responses.
* **Exploratory**: Best for gathering qualitative insights and exploring complex, open-ended topics.

### ****Difference Between Open-Ended and Closed-Ended Questions****

In interviews, the type of questions asked can significantly influence the type of data gathered. Here’s the distinction between **open-ended** and **closed-ended** questions:

#### **Open-Ended Questions**

**Characteristics:**

* **Invites Detailed Responses**: These questions allow interviewees to share their thoughts, opinions, and explanations in their own words.
* **Exploratory**: They encourage deep reflection and conversation, giving interviewees the freedom to elaborate.
* **Qualitative Data**: Responses are typically descriptive and provide more context.

**Examples:**

* "Can you describe the challenges you face when using this system?"
* "What features would you like to see in the new version?"
* "How do you envision the solution improving your current process?"

#### **Closed-Ended Questions**

**Characteristics:**

* **Limited Response Options**: These questions prompt the interviewee to choose from predefined responses (e.g., yes/no, multiple choice).
* **Quantitative Data**: Responses are generally easier to quantify and analyze statistically.
* **Specific Focus**: Closed-ended questions often focus on specific facts or clear answers.

**Examples:**

* "Do you use the system every day?" (Yes/No)
* "How satisfied are you with the product on a scale of 1-5?"
* "Would you prefer feature A over feature B?" (Yes/No)

**Key Differences Between Open-Ended and Closed-Ended Questions**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Open-Ended Questions** | **Closed-Ended Questions** |
| **Response Type** | Descriptive, narrative responses | Simple, fixed responses (e.g., yes/no, multiple choice) |
| **Data Type** | Qualitative (rich insights) | Quantitative (easier to analyze and compare) |
| **Flexibility** | High; allows for elaboration and new topics to arise | Low; limited to predefined options |
| **Time to Answer** | Longer, requires more thought and explanation | Shorter, quicker to answer |
| **Use Cases** | Exploration, understanding feelings or motivations | Facts, preferences, validation |

**Question 8:- Questionnaire Technique – Where we will use?**

A **questionnaire** is a research tool that consists of a set of questions designed to gather data from respondents in a structured format. This technique is particularly useful in business analysis for gathering both **quantitative** and **qualitative** data from a larger group of stakeholders or users. The responses to these questions help a Business Analyst (BA) to identify patterns, understand needs, validate assumptions, and collect feedback about products, services, processes, or systems.

A survey or questionnaire is used to elicit business analysis information including information about the customers, products, work practices, and attitudes from a group of people in a structured way and in relatively short period of time. Surveys are the preferred elicitation technique when faced with a large number of stakeholders or when stakeholders are geographically dispersed and you need to gather the same information from them.

**Examples**

1. How many times have you visited [website] in the past month?

None

Once

More than once

1. What is the primary reason for your visit to [website]?

 To make a purchase

To find more information before making a purchase in store

To contact customer services

1. Who did you purchase these products for ?

Self

Family Member

Friend

Colleague

On Behalf of business

Other

1. How much time do you spend using [product or service]?

* Less than a minute
* About 1 - 2 minutes
* Between 2 and 5 minutes
* More than 5 minutes

1. In the last month, what has been your biggest pain point?

* Finding enough time for important tasks
* Delegating work
* Having enough to do

1. What's your biggest priority right now?

* Finding a faster way to work
* Problem-solving
* Staff development

**Question 9 :-** How to Sort the Requirements – Where we will use?

Sorting requirements is a crucial step in managing and prioritizing tasks, especially in project management, product development, or business analysis. This helps focus on what's most important or time-sensitive and ensures the project meets its objectives efficiently

**Steps to Sort Requirements**

1. **Gather and Document Requirements**
   * Collect all requirements from stakeholders, customers, or team members.
   * Document them in a structured format like a spreadsheet or a requirements management tool.
2. **Categorize Requirements**
   * Group requirements by type (e.g., functional, non-functional, business, technical).
   * Categorize based on features, modules, or objectives.
3. **Define Sorting Criteria**
   * Common criteria include:
     + **Priority**: High, medium, low (based on business needs or stakeholder input).
     + **Value**: Impact on the business or customer satisfaction.
     + **Feasibility**: Technical and resource feasibility.
     + **Urgency**: Deadlines or dependencies.
     + **Risk**: Complexity or uncertainty involved.
4. **Use a Prioritization Method**  
   Apply a structured framework to sort and prioritize requirements:
   * **MoSCoW Method**: Classify as Must-have, Should-have, Could-have, Won’t-have.
   * **Kano Model**: Categorize by basic, performance, or excitement needs.
   * **Weighted Scoring**: Assign scores based on defined criteria.
   * **Cost-Benefit Analysis**: Evaluate cost vs. value delivered.
   * **ICE Scoring**: Rank based on Impact, Confidence, and Ease.
5. **Review and Validate**
   * Collaborate with stakeholders to validate the sorting.
   * Adjust based on new inputs or shifting priorities.
6. **Update Regularly**
   * Revise sorting as project scopes or external factors change.

**Where You Will Use This Process**

* **Project Management**:  
  To prioritize tasks or deliverables in a project timeline.
* **Product Development**:  
  To determine which features to build first or include in a Minimum Viable Product (MVP).
* **Business Analysis**:  
  To align requirements with organizational goals and resources.
* **Agile Methodologies**:  
  For creating and prioritizing backlog items in Scrum or Kanban.
* **Resource Allocation**:  
  To focus on high-impact requirements when resources are limited.
* **Strategic Planning**:  
  To align requirements with long-term goals and objectives.
* **Risk Management**:  
  To address high-risk or high-priority areas proactively.

We will sort the requirements in two ways such as functional requirements and Non-Functional requirements.

**Functional requirements** define a function that a system or system element must be qualified to perform and must be documented in different forms. The functional requirements describe the system's behavior as it correlates to its functionality.

**Examples of functional requirements** are authentication, business rules, audit tracking, certification requirements, transaction corrections, etc.

**Non-functional requirements** are not related to the software functional aspect. They can be the necessities that specify the criteria that can be used to decide the operation instead of specific behaviors of the system

**Examples** - usability, reliability, security, storage, cost, flexibility,conguration, performance, legal or regulatory requirements, etc

**Question 10 - Prioritise the Requirements – –Where we will use**

Large software systems have a few hundred to thousands of requirements. Neither are all requirements equal nor do the implementation teams have the resources to implement all the documented requirements. There are several constraints such as limited resources, budgetary constraints, time crunch, feasibility,etc., which brings in the need to prioritize requirements.

Most customers on their part have a reasonable idea of what they need and what they want. But during requirements elicitation the customer provides the Business Analyst (BA) with all the requirements that he feels will make his work easier. The customer is not wrong on his part; the BA needs to understand the needs of the business to prioritize the requirements

**Use a Prioritization Framework**

* Apply structured methods to rank requirements:
  + **MoSCoW Method**: Classify as Must-have, Should-have, Could-have, and Won’t-have.
  + **Kano Model**: Differentiate basic, performance, and excitement needs.
  + **Value vs. Effort Matrix**: Plot requirements based on their value and the effort required.
  + **Weighted Scoring**: Assign scores based on multiple criteria.
  + **100-Dollar Method**: Allocate "virtual currency" to prioritize requirements.

Most requirements are interdependent and you will hardly find any requirement that exists independently. To understand why we need a dependency map – let us take a scenario where you have 8 requirements X,Y,Z,P,Q,R,M,O and N with priorities, on a 5- level scale where 1 is most critical and 5 least critical, as1,2,1,4,5,1,2,2,3. So, with these priorities it would be logical to begin with requirements X, Z and R

2. MoSCoW – This prioritization technique was developed by DaiClegg of Oracle UK Consulting. it is one of the more widely used techniques for its simplicity and ease of use. The letters of the word MoSCoW stand for Must, Should, Could and Won’t.

Must have (or Minimum Usable Subset) – These are features that must be included before the product can be launched.

Should haves are features that are not critical for the launch, but are considered to be important and of a high value to the user.

Could haves are features that are nice to have and could potentially be included without incurring too much effort or cost

Won’t have - are features that have been requested but are explicitly excluded from scope for the planned duration and maybe included in a future phase of development.MoSCoW method works better than the numeric rating system as itis much easier for the stakeholders to rate the requirements as Must, Should, Could or Would.

**MUST (M)**

Defines a requirement that has to be satisfied for the final solution to be acceptable

**e.g**

. The HR system “must” store employee leave history.

**SHOULD (S)**

 This is a high-priority requirement that should be included if possible, within the delivery time frame. Workarounds may be available for such requirements and they are not usually considered as time-critical or must-haves.

**e.g**

. The HR system “should” allow the printing of leave letters.

**COULD (C)**

 This is a desirable or nice-to-have requirement (time and resources) but the solution will still be accepted if the functionality is not included

**e.g.**

 The HR system “could” send out notifications on pending leave dates.

**WON’T or WOULD (W)**

 This represents a requirement that stakeholders want to have, but have agreed will not be implemented in the current version of the system. That is, they have decided it will be postponed till the next next round of developments

**e.g**

. The HR system “won’t” support remote access but may do so in the next release

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

A weekly report is a document that gives an overview of the work an employee has completed in a given week. It provides a progress update on ongoing work and details the work planned for the following week

To effectively drive weekly status reporting, you should establish a consistent communication channel (like a dedicated project management tool), set clear expectations for what should be included in reports, encourage transparency by having team members regularly update their progress, hold regular meetings to discuss updates, and proactively address any roadblocks or issues raised in reports; ensuring everyone stays informed about project progress and can identify potential problems early on.

Key elements to include in a weekly status report:

* **Completed tasks:** A summary of what tasks were finished during the week.
* **Ongoing tasks:** Details on currently active tasks and their progress.
* **Upcoming tasks:** A list of planned tasks for the upcoming week.
* **Challenges and roadblocks:** Any obstacles encountered and potential solutions.
* **Key metrics:** Relevant project metrics or KPIs to track progress.
* **Dependencies:** Any dependencies on other team members or external factors.

How to drive weekly status reporting:

* **Set clear guidelines:**

Define the format of the report, due dates, and expected level of detail.

* **Utilize a project management tool:**

Leverage a platform like Asana, Trello, or Microsoft Teams to streamline reporting and facilitate easy access to updates.

* **Regular meetings:**

Hold weekly team meetings where individuals share their updates from their reports, allowing for discussion and collaborative problem-solving.

* **Leadership involvement:**

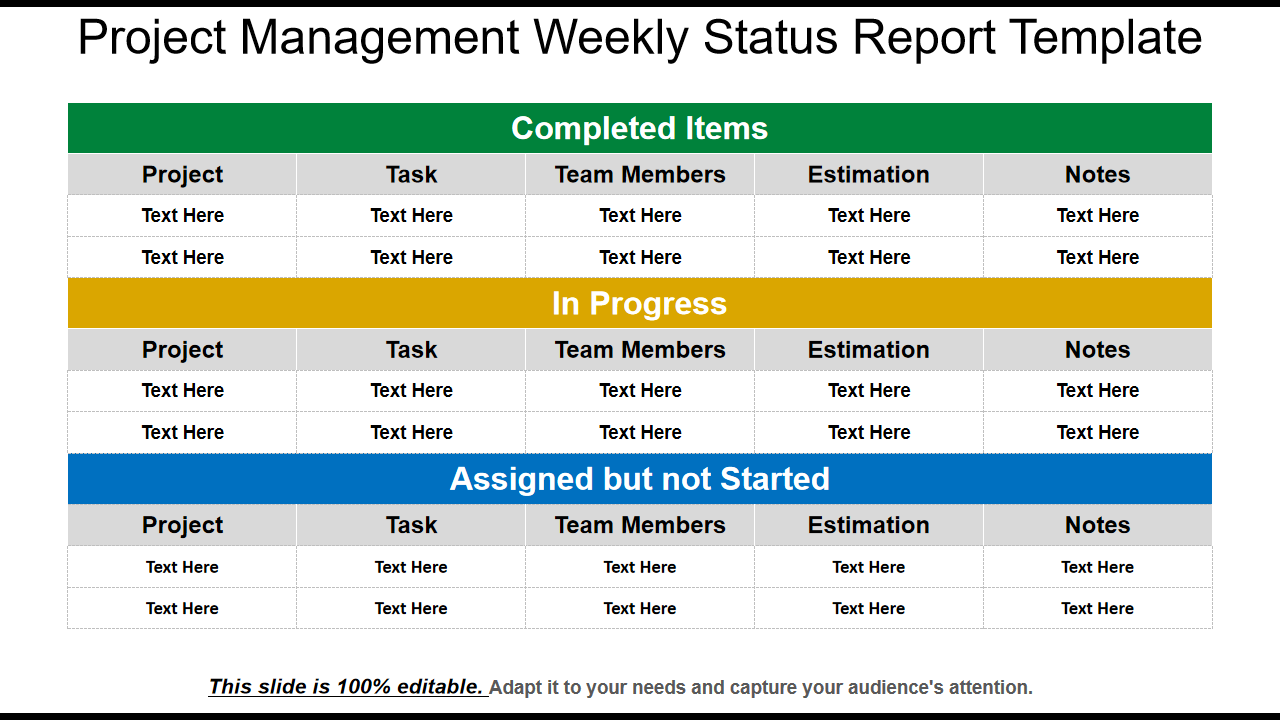
Encourage project managers and team leads to actively review reports and provide feedback.

* **Promote transparency:**

Foster a culture where team members feel comfortable raising concerns or reporting issues in their status updates

### ****Benefits of Weekly Status Reporting****

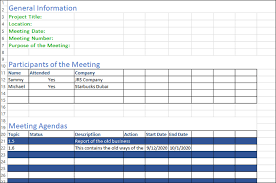
* **Transparency**: Keeps stakeholders informed about project progress.
* **Alignment**: Ensures everyone is on the same page regarding objectives and timelines.
* **Proactive Risk Management**: Identifies and addresses issues before they escalate.
* **Accountability**: Tracks team performance and ownership of tasks.
* **Decision-Making Support**: Provides data for timely and informed decisions.



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

Minutes is to create an official record of the actions taken at a Meeting. Minutes serve to both memorialize the actions taken for those attending the Meeting as well as for those who were unable to attend the Meeting.

Meeting minutes are notes that are recorded during a meeting. They highlight the key issues that are discussed, motions proposed or voted on, and activities to be undertaken

****

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

The role of BA in change request is very important as the change requests differ in number and complexity across business projects and may come in before, during or after implementation of a solution.

Below are the steps to follow—

>Understand the reason for the change—

>Understand the impact of the change—

>Understand the effort required to implement the change—

>Ensure that the change request follows the pre determined approval process

Change Tracker - Sample Document

Project Name: [Project Name]

Date: [Date]

Change ID: [Unique identifier for each change request]

Initiator: [Name of person requesting the change]

Description of Change:

* **Brief summary:** [Concise description of the proposed change]
* **Details:** [Detailed explanation of the change, including affected areas, functionalities, etc.]

Reason for Change:

* **Business need:** [Explanation of how the change aligns with business goals or addresses a problem]
* **Technical requirement:** [Technical reasons necessitating the change, if applicable]

Impact Analysis:

* **Scope:** [How the change affects the project scope]
* **Schedule:** [Potential impact on project timeline]
* **Cost:** [Estimated cost implications of the change]
* **Quality:** [Potential impact on quality standards]

Proposed Solution:

* **Implementation plan:** [Step-by-step approach to implementing the change]
* **Dependencies:** [Any other elements or tasks that rely on this change]

Approval Status:

* **Submitted to:** [Name of person reviewing the change]
* **Approval Date:** [Date of approval, if applicable]
* **Approved by:** [Signature or initials of approver]

Change Implementation:

* **Assigned to:** [Team member responsible for implementing the change]
* **Completion Date:** [Expected date of change implementation]

Change Verification:

* **Validation Method:** [How the change will be tested and verified]
* **Completed by:** [Name of person verifying the change]

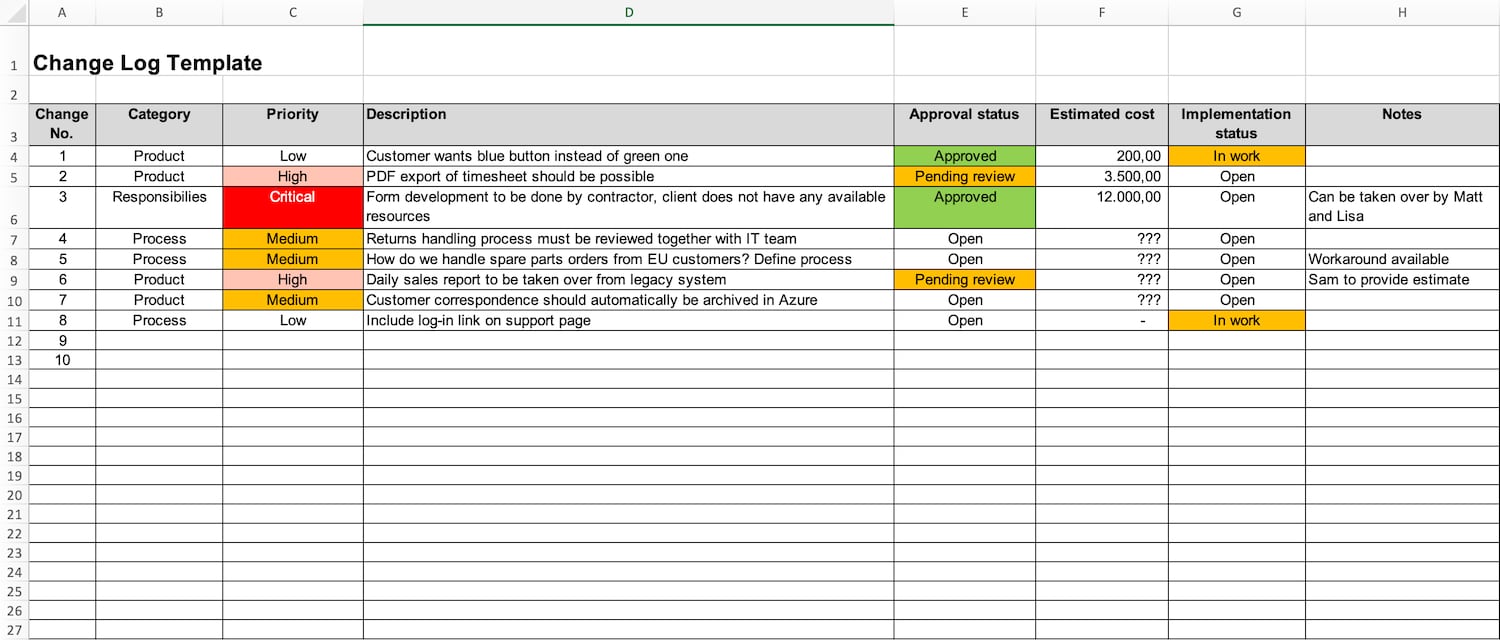
Change Closure:

* **Final Notes:** [Any additional information or observations about the change]

Revision History:

* **Date:** [Date of revision]
* **Changes Made:** [Description of modifications made to the change request]

****

****

**Q14. Difference between Traditional Development Model and Agile Development Models**

**A traditional development model**, often exemplified by the Waterfall method, follows a linear, rigid process with distinct phases where all requirements must be defined upfront,

**while an Agile development model** is iterative and flexible, allowing for continuous feedback and adaptation to changing needs throughout the project lifecycle, making it more responsive to evolving requirements and customer input; essentially,

**traditional models** prioritize planning and detailed documentation upfront, whereas

**Agile prioritizes collaboration** and quick feedback loops to deliver working software in short iterations

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Traditional Development Model** | **Agile Development Mode** |
| **Approach** | Sequential and linear (e.g., Waterfall model). | Iterative and incremental. |
| **Planning** | Heavy upfront planning; fixed requirements. | Adaptive planning; evolving requirements. |
| **Flexibility** | Low flexibility to changes once development begins. | High flexibility; accommodates changes throughout the process. |
| **Delivery** | Single delivery at the end of the project lifecycle. | Continuous delivery of functional increments. |
| **Feedback** | Limited feedback opportunities until the end. | Regular feedback from stakeholders and end-users. |
| **Team Structure** | Silos of specialized roles (e.g., developers, testers). | Cross-functional teams collaborating throughout. |
| **Customer Involvement** | Minimal involvement after the requirement phase. | Active involvement throughout the development process. |
| **Documentation** | Comprehensive documentation before development starts. | Lightweight documentation; emphasizes working software. |
| **Risk Management** | High risk as testing occurs late in the cycle. | Lower risk due to early and frequent testing. |
| **Tools and Practices** | Focus on traditional tools and processes. | Emphasis on collaboration tools (e.g., Jira, Scrum boards). |
| **Timeline** | Fixed timeline and milestones. | Adaptive timelines based on iterations or sprints. |

**Q15. Explain Brainstorming Technique – Where to use?**

### ****Brainstorming Technique****

**Definition**:  
Brainstorming is a creative group activity designed to generate a wide range of ideas or solutions to a specific problem or challenge. It encourages participants to think freely and share their thoughts without judgment, fostering innovation and out-of-the-box thinking.

### ****Key Principles of Brainstorming****

1. **Focus on Quantity**
   * The goal is to generate as many ideas as possible without worrying about quality initially.
2. **Withhold Criticism**
   * No idea is dismissed or criticized during the brainstorming session to ensure all participants feel safe to contribute.
3. **Welcome Unusual Ideas**
   * Creative and unconventional ideas are encouraged, as they often lead to innovative solutions.
4. **Build on Others' Ideas**
   * Encourage participants to build upon or combine ideas to create improved or new concepts.

### ****Steps to Conduct a Brainstorming Session****

1. **Define the Problem or Goal**
   * Clearly state the purpose of the session to give participants a focus.
2. **Assemble a Diverse Group**
   * Include individuals with varying expertise and perspectives to broaden the range of ideas.
3. **Set Ground Rules**
   * Outline rules such as withholding criticism and respecting others' contributions.
4. **Use Techniques to Spark Creativity**
   * Examples include mind mapping, reverse brainstorming, or SCAMPER (Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Rearrange).
5. **Record All Ideas**
   * Write down every idea on a whiteboard, sticky notes, or digital tools like Miro or MURAL.
6. **Refine and Evaluate**
   * After the brainstorming session, review and evaluate the ideas based on feasibility, impact, and alignment with the objectives.

### ****Where to Use Brainstorming****

1. **Problem-Solving**
   * When facing complex challenges requiring creative solutions (e.g., resolving customer complaints or improving processes).
2. **Product Development**
   * Generating ideas for new products, features, or enhancements.
3. **Strategic Planning**
   * Developing visions, goals, and strategies for long-term success.
4. **Marketing Campaigns**
   * Brainstorming slogans, advertisements, or promotional strategies.
5. **Innovation Projects**
   * Generating groundbreaking ideas or exploring new business opportunities.
6. **Team Building**
   * Encouraging collaboration and shared ownership in problem-solving.
7. **Workshops and Training**
   * Engaging participants in active learning through idea generation.

**You can use brainstorming throughout any design or work process, of course, to generate ideas for design solutions, but also any time you are trying to generate ideas, such as planning where to do empathy work, or thinking about product and services related to your project.**

Brain storming: It is a creative technique to find a solution or to understand the need or requirement by a group of people. As a BA, by using brainstorming, we can gather the ideas and can creative solutions for problems in short time. The steps involved in brainstorming

1. **Prepare for brainstorming**: start a clear and concise objective for the session. Generate as many ideas as possible and don’t limit the creative ideas instead limit the time for session. Decide who all are going to included in session and their role like participant or facilitator.

2. **Conduct brainstorming session**: Share new ideas without any discussion, criticism or evaluation. Record or note down all ideas.

3**. Wrap up the brainstorming**: once the time limit is reached create a list of ideas and eliminate the duplicates. Rate the ideas and prioritize the ideas using voting and distribute the final list of ideas.