BUSINESS ANALYST

REQUIREMENT GATHERING

Requirement Gathering is like creating a shopping list before cooking (*development phase*). We talk to everyone (*stakeholders*) involved, understand what they want, and make a detailed list of ingredients (*features*) needed. This ensures the final dish (*software*) meets everyone's tastes and needs, making the whole process smoother and more satisfying.

Techniques:

WORKSHOP

INTERVIEW

- **Understand the Purpose:** Know why you're gathering requirements, just like knowing why you're planning a party. It helps you focus on what's important.
- **Identify Key People**: Figure out who needs to be involved, like deciding which friends should come to your party. This ensures you don't miss anyone important.
- **Learn About the Project:** Get a basic idea of what the project is about, just like knowing if you're planning a birthday or a holiday party. It helps you ask the right questions.
- **Prepare Questions:** Think of questions to ask, like planning conversation starters for your party. This keeps the conversation going and helps gather useful information.
- **Be Open and Friendly:** Approach people with a positive attitude, just like being friendly at a gathering. It makes others more willing to share their thoughts and needs.
- **Have a System to Record Information (Must take consent):** Decide how you'll take notes, like choosing between using a notebook or your phone at a party. It helps you keep track of what people are saying.
- **Be Ready to Listen**: Be a good listener, just like paying attention when a friend is talking about their day. This helps you understand what people really want.

INTERVIEW TECHNIQUE

An **interview** is like a friendly chat to learn more about someone or something. It's a one-to-one conversation where you ask questions and listen to the answers. Just like getting to know a friend, interviews help understand people's thoughts and needs, important for making things work well together.

INTERVIEW PREPARATION:

1. **Identify Participants:**
   - Determine who the key stakeholders are, including end-users, business managers, subject matter experts, and technical experts.
2. **Define Objectives:**
   - Clearly outline the goals and objectives of the interview. What information are you seeking to gather? What decisions will be influenced by the interview?
3. **Develop an Interview Guide:**
   - Create a structured interview guide with a list of open-ended questions. Ensure that the questions are clear, concise, and tailored to the interviewee's role and expertise.
4. **Schedule Interviews:**
   - Coordinate with participants to schedule interviews at convenient times. Be mindful of time zones and participants' availability.

DURING THE INTERVIEW:

1. **Build Rapport:**
   - Begin the interview by introducing yourself and explaining the purpose of the interview. Establish a comfortable and open atmosphere to encourage communication.
2. **Open-Ended Questions:**
   - Use open-ended questions to elicit detailed responses and encourage stakeholders to express their thoughts. Examples:

    - *"Can you describe your role in the project and how you interact with the software/system?"*
    - *"What are the key challenges you currently face in your daily tasks that the new software should address?"*

3. **Probing for Details:**
    - Follow up on responses with probing questions to delve deeper into specific aspects. Examples:
        - *"Can you provide an example of a specific task where the current system falls short of your needs?"*
        - *"How do you envision the new system improving your workflow?"*
4. **Scenario-Based Questions:**
    - Present hypothetical scenarios to understand how stakeholders would handle specific situations. Examples:
        - *"Imagine a scenario where the system is down. How would that impact your work, and what measures would you take to mitigate the impact?"*
        - *"In a situation where you need to collaborate with a remote team member using the software, what features would be essential for you?"*
5. **Clarify Ambiguities:**
    - If an answer is unclear or ambiguous, seek clarification.
6. **Document Responses:**
    - Take detailed notes during the interview to capture key points, quotes, and any additional insights provided by the stakeholders.

AFTER THE INTERVIEW:

1. **Review and Validate:**
    - Review your notes to ensure accuracy and completeness. If necessary, follow up with participants for clarification or additional information.
2. **Identify Patterns and Themes:**
    - Analyse the interview data to identify common patterns, recurring themes, and discrepancies. Look for insights that can inform the development of requirements.

3. **Document Findings:**
    - Document the findings from the interviews in a clear and organized manner. Create a summary that highlights key takeaways and any potential conflicts or areas that require further exploration.

SAMPLE QUESTIONS (*Just for understanding*):

1. Stakeholder Identification:

- **Objective:** Identify and understand the various stakeholders involved in the project.
- **Sample Questions:**
    - Who are the primary users of the software?
    - Can you identify other stakeholders who will be affected by or have an impact on the project?

2. Current State Analysis:

- **Objective:** Understand the current processes, workflows, and pain points.
- **Sample Questions:**
    - *Can you describe the current process or workflow you follow for [specific task]?*
    - *What challenges or issues do you currently face in your daily activities related to the software?*

3. Functional Requirements:

- **Objective:** Gather information about the specific functionalities and features needed.
- **Sample Questions:**
    - *What are the essential features the software must have to meet your needs?*
    - *Can you provide examples of specific tasks that the software should be able to perform?*

4. Non-Functional Requirements:

- **Objective:** Explore aspects like performance, security, and usability requirements.
- **Sample Questions:**
    - *How fast do you expect the system to respond when performing [specific action]?*
    - *Are there any security or privacy concerns that need to be addressed?*

5. Data Requirements:

- **Objective:** Understand the data that will be input, processed, and output by the software.
- **Sample Questions:**
    - What types of data do you anticipate entering into the system?
    - How should the system handle and store sensitive information?

6. User Interface Preferences:

- **Objective:** Gather information about the user interface preferences and expectations.
- **Sample Questions:**
    - *What kind of user interface do you find most intuitive for your tasks?*
    - *Are there any specific design elements or styles you prefer or dislike?*

7. Integration Requirements:

- **Objective:** Identify any external systems or tools that need to integrate with the software.
- **Sample Questions:**
    - *Are there any existing systems or tools that the new software needs to integrate with?*
    - *How should data be exchanged between the new system and other existing systems?*

8. Change Management:

- **Objective:** Understand how stakeholders expect changes to be managed and communicated.
- **Sample Questions:**
    - *How do you prefer to be informed about changes or updates to the software?*
    - *What challenges do you foresee in adopting a new software solution?*

9. Training and Support:

- **Objective:** Identify training and support requirements for users.
- **Sample Questions:**
    - *What level of training do you think is necessary for users to effectively use the software?*
    - *How would you like to access support if you encounter issues with the software?*

10. Future Enhancements:

- **Objective:** Explore potential future features or improvements.
- **Sample Questions:**
    - *Can you foresee any changes in your workflow that might require additional features in the future?*
    - *What would you consider as valuable enhancements to the software over time?*

WORKSHOP TECHNIQUE

A **workshop** is like a team meeting where everyone (stakeholders) gets together to discuss and work on a project. It's a hands-on, interactive session where people share ideas, solve problems, and make decisions. It's like a collaborative board meeting where everyone brings something to contribute and leaves with a plan.

WORKSHOP PREPARATION:

1. **Define Workshop Objectives:**
    - Clearly outline the goals and objectives of the workshop. What specific outcomes do you aim to achieve? Examples include identifying key features, prioritizing requirements, or resolving conflicting perspectives.
2. **Select Participants:**
    - Identify and invite key stakeholders representing various perspectives and expertise relevant to the project.
3. **Prepare Materials:**
    - Develop materials such as presentation slides, flip charts, sticky notes, and markers to facilitate activities during the workshop.
4. **Create an Agenda:**
    - Outline a detailed agenda with specific activities, time allocations, and breaks. Clearly communicate the purpose of each session.
5. **Mail Questionnaires:**
    - Prepare set of questionnaires and mail it to stakeholders so that they can be prepared beforehand.

DURING THE WORKSHOP:

1. **Icebreaker Activity:**
    - Start with a brief icebreaker to help participants feel more comfortable and encourage interaction.
2. **Introduction and Context Setting:**
    - Provide a brief overview of the project, its goals, and the importance of the workshop. Clarify the roles and expectations of participants.
3. **Brainstorming Sessions:**
    - Use brainstorming techniques to generate ideas related to the project. Sample questions:
        - *"What are the must-have features for the new software?"*
        - *"In an ideal scenario, how would the software improve your daily tasks?"*
4. **Affinity Diagrams:**
    - Organize ideas generated during brainstorming into categories using affinity diagrams. Sample questions:
        - *"Based on our discussion, can we group these ideas into common themes or categories?"*
        - *"Are there any overlaps or dependencies between the ideas we've identified?"*
5. **Prioritization Activities:**
    - Engage participants in activities to prioritize features or requirements. Sample questions:
        - *"Considering limited resources, what features do you believe should be prioritized for the initial release?"*
        - *"What criteria should we use to determine the priority of these features?"*
6. **Use Case Development:**
    - Work with participants to develop detailed use cases or user stories. Sample questions:
        - *"Can you describe a specific scenario where the software would be used, including the steps involved and the desired outcome?"*

        - *"What are the alternative paths or exceptions we should consider in this use case?"*

7. **Prototype Feedback:**

    - If applicable, present a prototype or wireframes of the software and gather feedback. Sample questions:

        - *"How does the proposed user interface align with your expectations and workflow?"*

        - *"Are there any additional features or changes you would like to see in the prototype?"*

8. **Group Discussions:**

    - Facilitate group discussions to address any conflicts or differing opinions. Sample questions:

        - *"We've heard different perspectives on [specific issue]. How can we find a resolution or compromise that meets everyone's needs?"*

        - *"Are there any concerns or risks that have not been adequately addressed?"*

AFTER THE WORKSHOP:

1. **Documentation:**

    - Document the outcomes of the workshop, including key insights, decisions, and unresolved issues.

2. **Share Findings:**

    - Communicate the workshop findings with stakeholders who couldn't attend and validate the documented information with participants.

3. **Action Items:**

    - Identify and assign action items based on the workshop outcomes. Clearly communicate responsibilities and deadlines.

4. **Follow-Up:**

    - Schedule follow-up sessions or individual meetings to address any outstanding issues and ensure ongoing alignment with stakeholders.