(1) Interviews

Interviews in requirements elicitation represent a fundamental, direct method for gathering detailed information from stakeholders. It involves structured or semi-structured one-on-one or group discussions with individuals who have a stake in the project, such as end-users, business managers, project sponsors, and others who possess insights into the system's requirements. Through interviews, requirements analysts can delve deeply into the stakeholders' needs, expectations, and experiences, facilitating a thorough understanding of the requirements for the new or improved system.

Key Objectives and Advantages:

- **Depth of Insight:** Interviews provide an opportunity to explore complex issues in detail, allowing for a deeper understanding of stakeholder needs and the nuances of their requirements.
- **Clarification and Verification:** They offer a direct channel for clarifying ambiguities and verifying assumptions, ensuring the elicited requirements are accurate and fully understood.
- **Flexibility:** The format of interviews can be adapted to suit the stakeholder's familiarity with the subject matter and the specific goals of the elicitation process, ranging from open-ended discussions to more structured question-and-answer formats.
- **Personal Engagement:** Interviews facilitate personal interaction, building trust and rapport with stakeholders, which can encourage openness and sharing of critical insights that might not emerge through other elicitation techniques.

Process and Implementation:

1. **Planning:** Identify the stakeholders to be interviewed and the objectives for each interview. Prepare a list of questions or topics to be covered, tailored to the interviewee's role and level of expertise.
2. **Conducting Interviews:** Depending on the chosen format (structured, semi-structured, or unstructured), the interviewer guides the conversation through prepared questions or topics while remaining open to exploring new insights that emerge.
3. **Active Listening:** It's crucial for the interviewer to practice active listening, paying close attention to the interviewee's responses and asking follow-up questions to probe deeper into key areas.
4. **Documentation:** Detailed notes or recordings (with the interviewee's consent) should be taken to ensure that all information is captured accurately for later analysis.
5. **Analysis:** The collected data is analyzed after the interview to identify and document the requirements. This may involve coding responses, identifying themes, and prioritizing the requirements based on the information gathered.

Challenges and Considerations:

- **Bias and Influence:** Interviewers must be aware of potential biases and strive to maintain neutrality, ensuring that the interviewee's responses are not unduly influenced by how questions are phrased or presented.
- **Time and Resource Intensive:** Conducting and analyzing interviews can be time-consuming, particularly for projects with many stakeholders. Efficient planning and prioritization of interviews are essential.
- **Interpretation and Accuracy:** The subjective nature of personal communication requires careful interpretation of responses, particularly for open-ended questions, to ensure that the requirements are accurately understood and documented.

(2) Surveys/Questionnaires

Surveys and questionnaires stand as highly scalable and efficient techniques for requirement elicitation, enabling data collection from a broad audience in a relatively short period of time. It is particularly useful when the project stakeholders are numerous or geographically dispersed, and there's a need to gather a wide range of opinions, preferences, and requirements for the system under development. It facilitates the quantitative and qualitative stakeholder needs analysis by deploying structured questions.

Key Objectives and Advantages:

- **Broad Reach:** Surveys and questionnaires can be distributed to many stakeholders simultaneously, making it possible to gather diverse perspectives efficiently.
- **Quantitative and Qualitative Data:** They can be designed to collect quantitative data (e.g., ratings, rankings) and qualitative insights (e.g., open-ended responses), providing a balanced view of stakeholder requirements.
- **Anonymity and Honesty:** Respondents might be more willing to provide honest feedback when anonymity is assured, leading to more accurate and truthful responses.
- **Cost-Effective:** Compared to other elicitation methods such as interviews and workshops, surveys and questionnaires are more cost-effective, especially when stakeholders are widespread.

Process and Implementation:

1. **Designing the Survey/Questionnaire:** Carefully craft questions that align with the objectives of the requirements elicitation. The survey should include a mix of closed-ended questions for statistical analysis and open-ended questions to capture detailed comments and suggestions.
2. **Pilot Testing:** Before widespread distribution, conduct a pilot test with a small, representative segment of the target audience to identify any ambiguities or issues in the questionnaire.
3. **Distribution:** Choose the most effective means to distribute the survey, considering the stakeholders' access to and familiarity with digital tools. In some settings, options include email, online survey platforms, or even paper-based questionnaires.
4. **Data Collection:** Set a reasonable response deadline, and consider sending reminders to maximize the response rate.
5. **Analysis:** Analyze the collected data to identify trends, patterns, and outliers. Quantitative data can be statistically analyzed, while qualitative responses require content analysis to extract meaningful insights.
6. **Feedback and Validation:** Share the findings with key stakeholders for validation and to ensure that the interpreted requirements accurately reflect their needs and expectations.

Challenges and Considerations:

- **Design Complexity:** Crafting clear, unbiased questions capable of eliciting useful information requires careful consideration and expertise in survey design.
- **Response Rate and Bias:** Achieving a high response rate can be challenging, and the results may be biased toward the views of those who chose to respond.
- **Interpretation of Responses:** Analyzing open-ended responses and translating them into actionable requirements necessitates a deep understanding of the context and the ability to interpret stakeholder feedback accurately.

Question Types:

- **Closed-Ended Questions:** These questions limit responses to a set of predefined options. They are useful for gathering quantitative data that can be easily analyzed. Examples include multiple-choice questions, Likert scale questions for assessing attitudes or preferences, and yes/no questions.
- **Open-Ended Questions:** These allow respondents to answer in their own words, providing qualitative insights that can reveal nuanced understanding and novel ideas. While they are more valuable, they require more effort to analyze.
- **Ranking and Rating Questions:** These questions ask respondents to prioritize or rate different items according to their preferences or importance. They are useful for understanding the relative significance of various requirements.

(3) Brainstorming

Brainstorming, as a requirements elicitation technique, embodies a dynamic group activity focused on generating a wide array of ideas, solutions, and requirements for a project. It thrives on leveraging the collective intelligence and creativity of the participants, usually comprising project stakeholders, team members, and potential users. This technique is especially valuable in the initial phases of a project, where the goal is to explore various possibilities and identify innovative solutions without the constraints of criticism or feasibility considerations.

Key Objectives and Advantages:

- **Idea Generation:** Facilitates the rapid generation of a broad spectrum of ideas, allowing teams to explore various possibilities that might not emerge through individual contemplation.
- **Enhanced Collaboration:** Encourages active participation from all stakeholders, fosters a sense of ownership and collaboration across the project team, and ensures a diverse set of perspectives is considered.
- **Creative Freedom:** Creates a safe space for free thinking and sharing out-of-the-box ideas, which can lead to innovative solutions and uncover hidden requirements.
- **Problem-Solving:** Helps identify and solve complex problems by allowing team members to build on each other's ideas, leading to more refined and comprehensive solutions.

Process and Implementation:

1. **Preparation:** Define the scope and objectives of the brainstorming session, select a diverse group of participants, and choose a facilitator to guide the process.
2. **Idea Generation Phase:** Participants are encouraged to freely express their ideas, no matter how unconventional they may seem, without fear of immediate critique or evaluation.
3. **Encouragement of Diverse Ideas:** The facilitator encourages the exploration of different angles and perspectives, ensuring a wide-ranging discussion that can lead to innovative solutions.
4. **Building on Ideas:** Participants build on each other's suggestions, enhancing and expanding upon initial concepts, often leading to more refined and creative outcomes.
5. **Documentation:** All ideas are recorded verbatim, ensuring nothing is lost or overlooked during the session. This record serves as a valuable resource for subsequent analysis and development phases.
6. **Analysis and Refinement:** Following the session, ideas are categorized, evaluated, and refined. This stage may involve prioritization techniques to identify the most promising or critical ideas for further exploration or development.

Challenges and Considerations:

- **Group Dynamics:** Managing group dynamics to ensure equal participation and prevent dominance by more vocal participants is crucial for the success of a brainstorming session.
- **Idea Saturation:** There may be points during the session where ideas start to wane; the facilitator must employ strategies to reinvigorate the group and stimulate further creativity.
- **Quality vs. Quantity:** While brainstorming emphasizes the quantity of ideas over quality, it's essential to eventually shift focus towards filtering and refining ideas to ensure they align with project goals and constraints.

(4) Prototyping

Prototyping is a dynamic and interactive requirements elicitation technique that involves creating preliminary versions of a software system to explore ideas, uncover requirements, and gather feedback from users and stakeholders. It allows for a tangible exploration of the system's functionality and design before developing the full system. Prototyping bridges the initial concept and the final product, facilitating a deeper understanding and communication among developers, users, and stakeholders. Here's an in-depth look at how prototyping functions within the context of requirement elicitation:

Purpose and Benefits:

- **Visualization and Concretization:** Prototyping converts abstract requirements into tangible forms, enabling stakeholders to interact with a proposed system's elements. This visualization helps clarify, refine, and validate requirements.
- **Feedback Loop:** It creates a continuous feedback loop, allowing users to provide immediate and actionable insights. This iterative process helps identify misunderstandings or missing requirements early in the development cycle.
- **Experimentation and Exploration:** Developers and stakeholders can experiment with different approaches and designs to explore the feasibility of certain features or requirements. It encourages innovation and creative solutions.

Types of Prototypes:

1. **Low-Fidelity Prototypes:** These are quick and easy to create, often using paper sketches or simple digital mockups. They are useful for initial brainstorming and concept discussions.
2. **High-Fidelity Prototypes:** More sophisticated and closer to the final product, these prototypes offer interactive features and a detailed user interface representation. They are used for more detailed feedback and usability testing.
3. **Functional Prototypes:** These include working software elements, focusing on functional aspects rather than detailed design. They help in understanding the technical feasibility and functional behavior of the system.

Process and Implementation:

- **Identify Prototyping Goals:** Clearly define what aspects of the system the prototype will explore, such as specific functionalities, user interfaces, or workflows.
- **Develop the Prototype:** Create the prototype using appropriate tools and technologies based on the goals. The complexity of the prototype can vary depending on the requirements and the stage of the elicitation process.
- **Gather Feedback:** Present the prototype to users and stakeholders, encouraging them to interact with it and provide feedback on its functionality, design, and usability.
- **Iterate and Refine:** Use the feedback to revise and enhance the prototype. This iterative process may involve several rounds of prototyping and feedback to converge on the final set of requirements.

Challenges and Considerations:

- **Managing Expectations:** Ensure that stakeholders understand the purpose of the prototype and do not mistake it for the final product. Clear communication about the scope and objectives of prototyping is crucial.
- **Resource Allocation:** While prototyping can save time and resources in the long run by preventing rework, it does require an initial investment of time and resources. Balancing the depth and detail of prototyping against available resources is essential.
- **Integration with Other Techniques:** Prototyping is often most effective when used in conjunction with other requirement elicitation techniques, such as interviews, surveys, and workshops. This multi-faceted approach ensures a comprehensive understanding of requirements.

(5) Use Case Approach

The Use Case Approach in requirements elicitation is a method that focuses on identifying and defining the interactions between a user (or "actor") and a system to achieve specific goals. It helps in capturing functional requirements by describing how the system should behave from the user's perspective, providing a clear and concise way to communicate system behavior to both technical and non-technical stakeholders. It plays a crucial role in the early phases of software development, ensuring that the software functionality aligns with user needs and expectations.

Key Components of the Use Case Approach:

- **Actors:** Represents the users or other systems interacting with the subject system. Actors are external entities that initiate an interaction with the system to accomplish a goal.
- **Use Cases:** Describes a sequence of actions the system performs that yields an observable value result to an actor. A use case is a specific situation or scenario under which the system interacts with its environment.
- **Scenarios:** Detailed narratives or sequences of events, including main, alternative, and exceptional flows, illustrating how actors interact with the system across different use cases.

Process and Implementation:

1. **Identify Actors:** Identify all potential system users and other systems that might interact with it. This includes direct users, indirect users, and external systems.
2. **Define Use Cases:** For each actor, define the specific interactions they have with the system. This includes the main objectives or tasks the actor wants to accomplish using the system.
3. **Write Scenarios:** For each use case, write detailed scenarios that describe the steps the actor and the system take to achieve the goal. This includes the ideal path (main scenario) and variations (alternative and exception scenarios).
4. **Prioritize Use Cases:** Prioritize the use cases based on factors such as business value, frequency of use, and complexity. This helps focus development efforts on the most critical aspects of the system.
5. **Validation and Refinement:** Validate the use cases and scenarios with stakeholders to ensure they accurately represent user requirements. Refine the use cases based on feedback.

Benefits of the Use Case Approach:

- **User-Centric:** Focuses on user interactions, ensuring the system meets the actual needs and expectations of its users.
- **Clear Communication:** Provides a common language for discussing system requirements among stakeholders, including non-technical users.
- **Identification of Functional Requirements:** Helps in systematically identifying all the functional requirements of a system through the exploration of various user interactions.
- **Facilitates Testing and Validation:** Use cases can be directly used as a basis for developing test cases and validation criteria.

Challenges:

- **Complexity in Large Systems:** Managing and maintaining the use cases can become challenging for systems with many use cases.
- **Overlooking Non-Functional Requirements:** While excellent for capturing functional requirements, the use case approach may overlook non-functional requirements unless explicitly addressed.

The Use Case Approach in requirements elicitation is a powerful tool for understanding and documenting how a system should interact with its users. Focusing on the user's goals and describing system interactions from the user's perspective ensures that the developed system can perform its intended functions effectively and meet user expectations.

(6) User observation

In **user observation**, the team observes how users interact with existing systems to identify unspoken needs or problems.

Types of approaches:

- In **passive observation**, the analysis is carried out without the direct involvement of the observer. The observation can be carried out by recording video cameras and surveillance camera.
- In **active observation**, the analysis is carried out with the direct involvement of the observer. He can ask questions.
- In **explanatory observation**, the user explains what they are doing while using the product. The observer takes notes for analysis using the explanation given by the user.

Process and Implementation:

1. **Plan the observation**: first, define the objectives of your observation and how it aligns with your project goals and scope. Additionally, identify who you want to observe and how to recruit or access them, considering the diversity, representativeness and availability of your sample. Moreover, decide where and when users will be observed, considering the naturalness, relevance and accessibility of the environment. Lastly, choose the tools and techniques used to observe, record and analyze data, making sure they are reliable, valid and usable.
2. **Conducting observation**: follow best practices to ensure the quality and accuracy of your data. This includes obtaining consent from users and stakeholders, informing them of the purpose and scope of observation and protecting their privacy and confidentiality. Be flexible and responsive to changing conditions, focusing on relevant and significant aspects of user behavior and context, while avoiding distractions and assumptions. Document your observations using notes, audio, video, photos or other means in a consistent and clear format.
3. **Analyzing observation**: After you conduct observation, analyze your data and extract meaningful and actionable insights for your requirements. Combining and comparing your data with other sources of information can be assisted by matrices, maps, scenarios or stories. Lastly, presenting and sharing your findings and recommendations with your team and stakeholders can be done through reports, dashboards, prototypes or mockups.

Challenges and Considerations:

- **Time and Resource Intensive:** Conducting and analyzing observations can be time-consuming, particularly for projects with many stakeholders. Efficient planning and prioritization of observations are essential.
- **Interpretation and Accuracy:** Observations can be incomplete (especially for processes that are executed over a long time).

(7) Document analysis

Document analysis is the process of reviewing and examining documents, such as business plans, policies, contracts, manuals, reports, diagrams, or code, to identify relevant information and requirements for a project. Document analysis can help you understand the current state, the desired state, the gaps, the constraints, and the assumptions of the stakeholders and the system. Document analysis can also reveal implicit and hidden requirements that are not explicitly stated or communicated, but are implied or expected by the stakeholders or the system.

Process and Implementation:

1. **Plan and prepare.** Identify the purpose and scope of your document analysis, the types of documents to review, the sources and locations of the documents, the criteria and methods for selecting and evaluating the documents, and the tools and techniques for recording and organizing the information. Obtain the permission and access to the documents, and schedule enough time and resources for the document analysis.
2. **Conduct the analysis with a systematic and structured approach**: collect the documents that are relevant and reliable for your project, and sort them according to their categories, types, and formats. Read and analyze the documents carefully, and look for information and requirements that are related to your project scope and objectives. Use techniques such as highlighting, annotating, summarizing, cross-referencing, or tracing to extract and document the information and requirements Verify and validate the information and requirements with the stakeholders and the system, and resolve any conflicts or ambiguities.
3. **Analyze the results**: Synthesize and integrate the information and requirements from different documents, and identify the commonalities, differences, dependencies, and gaps. Classify and prioritize the information and requirements according to their types, levels, and importance. Evaluate and refine the information and requirements to ensure they are clear, complete, consistent, feasible, testable, and traceable. Communicate and review the information and requirements with the stakeholders and the system, and obtain their feedback and approval.

Challenges and Considerations:

- **Quality of resources**: Consider factors such as the availability and quality of the documents, the complexity and volatility of the requirements, the number and diversity of the stakeholders, the time and cost of the elicitation, and the level of detail and accuracy of the elicitation.

(8) Focus Group and workshops

By using a focus group, you can get information about a product, service from a group. The Focus group includes subject matter experts. The objective of this group is to discuss the topic and provide information. A moderator manages this session. This moderator should work with business analysts to analyze the results and provide findings to the stakeholders.

If a product is under development and the discussion is required on that product then the result will be to update the existing requirement or you might get new requirements. If a product is ready to ship then the discussion will be on releasing the product.

A Focus group or a workshop **is not an interview session** conducted as a group; rather it is a discussion during which feedback is collected on a specific subject. The session results are usually analyzed and reported. A focus group typically consists of 6 to 12 members. If you want more participants then create more than one focus group.

Benefits of the Use Case Approach:

- You can get information in a single session rather than conducting one to one interview.
- Active discussion with the participants creates a healthy environment.
- One can learn from other's experiences.

Challenges and Considerations:

- It might be difficult to gather the group on the same date and time.
- If you are doing this using the online method then the participant's interaction will be limited.
- A Skilled Moderator is required to manage focus group discussions.

(9) Introspection

Introspection is another elicitation technique where the requirement analyst “imagines” what kind of system is required for doing the required job, or by using available equipment. This technique is used when the end user is not available, not willing to answer the question or shows lack of feedback. In this technique the analyst imagines what requirement is needed by posing questions to himself (mirroring) to determine the requirements for the system.

It is not considered as a very good approach, because the analyst projects his needs. If you consider this method, apply it only at the very start of the project. Use others methods to complete the requirement analysis.

References

1. S. Tariq, A. Ibrahim, A. Usama and M. S. Shahbaz, "An Overview of Requirements Elicitation Techniques in Software Engineering with a focus on Agile Development," 2021 4th International Conference on Computing & Information Sciences (ICCIS), Karachi, Pakistan, 2021, pp. 1-6, doi: 10.1109/ICCIS54243.2021.9676192.
2. https://www.testbytes.net/blog/requirements-elicitation/
3. https://www.linkedin.com/advice/3/how-do-you-use-document-analysis-elicit
4. https://ebooks.inflibnet.ac.in/csp8/chapter/requirements-elicitation-techniques/
5. https://www.softwaretestinghelp.com/requirements-elicitation-techniques/