

1. Waterfall Methodology

The Waterfall Methodology is a linear and sequential project management approach where each phase such as requirements, design, development, testing, and deployment must be completed before moving to the next. It relies on detailed upfront planning and documentation, making it difficult to accommodate changes once a phase is finished. This model provides a structured framework with clear milestones and deliverables, making it best suited for projects with well-defined, stable requirements and predictable outcomes.

2. Requirement Gathering

Requirement gathering is the process of identifying, analyzing, and documenting the needs and expectations of stakeholders for a project or system. It helps ensure that the final product meets business objectives and user requirements. This phase involves interviews, surveys, meetings, and document analysis. Accurate requirement gathering lays the foundation for successful project development and delivery.

3. Requirement Analysis

Requirement analysis is the process of examining and refining gathered requirements to ensure they are clear, complete, and feasible. It involves identifying conflicts, dependencies, and priorities among requirements. This phase helps define what the system should do and how it should perform. Effective requirement analysis ensures a solid understanding before design and development begin.

4. Elicitation Technique

Requirement elicitation techniques are methods used to gather information from stakeholders to understand their needs and expectations for a system. Common techniques include interviews, questionnaires, workshops, brainstorming sessions, and document analysis. These techniques help uncover explicit and hidden requirements. Effective elicitation ensures that all stakeholder perspectives are captured accurately for project success.

5. Brainstorming

Brainstorming is a requirement elicitation technique used to generate a wide range of ideas and solutions through group discussion. It encourages open communication and creativity without immediate criticism or evaluation. Participants freely share their thoughts to explore multiple possibilities. This technique helps uncover innovative ideas and clarify project requirements effectively.

6. Document Analysis

Document analysis is a requirement elicitation technique that involves reviewing existing documents to understand current processes, system functionalities, and business rules. It helps identify gaps, improvements, and detailed requirements for the new system. Common documents analyzed include reports, manuals, and process maps. This technique provides valuable background information and supports accurate requirement definition.

7. Reverse Engineering

Reverse engineering is a requirement elicitation technique used to analyze an existing system to understand its components, design, and functionality. It helps identify system requirements when documentation is missing or outdated. By deconstructing the system, analysts can determine how it works and what improvements are needed. This technique is especially useful for system upgrades or replacements.

8. Focus Group

A focus group is a requirement elicitation technique that gathers a small group of stakeholders or users to discuss their needs, opinions, and expectations about a system. It encourages interactive discussions to uncover collective insights and preferences. The facilitator guides the conversation to explore specific topics or features. This technique helps capture diverse viewpoints and refine requirements effectively.

9. Prioritize Requirements

Prioritizing requirements is the process of ranking project requirements based on their importance, value, and impact on business goals. It helps teams decide which features should be developed first and which can be delayed or omitted. Factors like cost, time, risk, and stakeholder needs are considered. This process ensures optimal resource use and delivers maximum business value.

10. MoSCoW

MoSCoW is a requirement prioritization technique that categorizes requirements into four groups: **Must have**, **Should have**, **Could have**, and **Won't have**. It helps stakeholders clearly understand the importance and urgency of each requirement. This method ensures that critical features are delivered first within project constraints. MoSCoW promotes better decision-making and efficient project planning.

11. 100 Dollars Test

The 100 Dollars Test is a requirement prioritization technique where stakeholders are given a virtual budget of 100 dollars to distribute among different requirements. Each stakeholder allocates the amount based on the importance of each feature or function. The total distribution helps identify which requirements hold the highest value. This method encourages collaboration and provides a clear, quantitative view of priorities.

12. Validating Requirements

Validating requirements is the process of ensuring that documented requirements accurately reflect stakeholder needs and align with business objectives. It verifies that each requirement is clear, complete, feasible, and testable. Techniques like reviews, walkthroughs, and prototyping are commonly used. Effective validation helps prevent errors and costly changes during later project stages.

13. FURPS

FURPS is a model used to classify software quality attributes into five categories: **Functionality, Usability, Reliability, Performance, and Supportability**. It helps evaluate both functional and non-functional requirements of a system. This model ensures a balanced focus on user needs and system performance. FURPS guides teams in delivering high-quality and maintainable software solutions.

14. SMART

SMART is a framework used to define clear and achievable project or requirement goals. It stands for **Specific, Measurable, Achievable, Relevant, and Time-bound**. This approach ensures that each requirement is well-defined and trackable throughout the project. Applying SMART criteria helps improve focus, accountability, and project success.

15. Agile Methodology

Agile Methodology is an iterative and flexible approach to software development that focuses on delivering small, usable parts of a project in short cycles called sprints. It promotes continuous collaboration between teams and stakeholders. Changes and feedback are welcomed at any stage to improve the product. Agile ensures faster delivery, adaptability, and higher customer satisfaction.

16. SCRUM

Scrum is an Agile framework that organizes work into short, time-boxed iterations called sprints, usually lasting two to four weeks. It focuses on teamwork, accountability, and continuous improvement. Key roles include the Product Owner, Scrum Master, and Development Team. Scrum helps deliver valuable, high-quality increments of a product efficiently and adaptively.

17. User story

A user story is a simple, short description of a feature told from the end user's perspective. It typically follows the format "As a [user], I want [goal] so that [benefit]." User stories help teams understand what the user needs and why. They promote collaboration and ensure the development delivers real user value.

18. Product Backlog

A product backlog is a prioritized list of all features, enhancements, and fixes required for a product. It is maintained by the Product Owner and serves as the single source of work for the Scrum team. Items in the backlog are continuously refined and reprioritized based on feedback and changing needs. The backlog ensures that the most valuable tasks are addressed first during development.

19. Sprint Backlog

A sprint backlog is a list of selected product backlog items that the Scrum team commits to completing during a specific sprint. It includes user stories, tasks, and goals planned for that iteration. The sprint backlog is owned and managed by the development team. It provides a clear focus on what will be delivered within the sprint timeframe.

20. Sprint Planning

Sprint planning is a Scrum ceremony where the team decides what work will be completed during the upcoming sprint. The Product Owner presents the prioritized backlog items, and the team selects tasks they can commit to. Goals and deliverables for the sprint are clearly defined. This meeting ensures everyone understands the work scope and objectives before development begins.

21. Scrum Meeting

A Scrum meeting, also known as the daily stand-up, is a short daily session where the team discusses progress and plans for the day. Each member shares what they did yesterday, what they will do today, and any obstacles faced. It helps maintain transparency and team coordination. The meeting ensures continuous alignment toward sprint goals.

22. Sprint retrospective meeting

A sprint retrospective meeting is held at the end of each sprint to reflect on the team's performance and processes. The team discusses what went well, what didn't, and how improvements can be made. It promotes continuous learning and teamwork. The goal is to enhance efficiency and effectiveness in future sprints.

23. Sprint Review meeting

A sprint review meeting is conducted at the end of each sprint to showcase the completed work to stakeholders. The Scrum team demonstrates the product increment and gathers feedback. This meeting encourages collaboration and transparency about progress. It helps ensure the product is evolving in line with business and user needs.