Name: Snehal Wankhade  
Topic: Forums20

1. **Requirement Gathering**: The process of collecting, documenting and managing business needs from stakeholders to ensure project’s success. Business analysts use techniques like interviews, focus group, surveys to understand stakeholder’s requirements. Proper gathering ensures alignment with organizational goals and reduces the risk of project failure.
2. **Stakeholder Analysis**: A structured approach to identify and analyze individual or groups affected by a project. It assesses their influence, expectations and communication needs to ensure their perspectives are considered in decision making. Stakeholder maps and engagement strategies are tools often used.
3. **Gap Analysis**: This technique compares the current state of processes, systems or operations to desired future state. It identifies gaps that need addressing to meet objectives, providing actionable insights for bridging those gaps. Commonly used for process improvement or system upgrades.
4. **Use Cases**: Use cases describes interaction between a user (actor) and system to achieve a specific goal. They include actors, triggers, steps, outcomes, providing a detailed understanding of system functionality. These are essential for designing user-centered solutions.
5. **Business Process Model**: A visual representation of workflows and processes using notations like UML. It helps to identify inefficiencies, redundancies, and opportunities for improvement. Business Analysts use these models to communicate current and proposed processes to stakeholders.
6. **Waterfall Model**: A linear and sequential project management approach where each phase requirement gathering, analysis, design, development, testing is completed before moving to the next. It’s ideal for projects with well-defined goals and minimal scope for changes.
7. **Agile Methodology**: A project management approach emphasizing iterative progress, collaboration and flexibility. Agile breaks projects into smaller increments (sprints) and encourage constant feedback from stakeholders. Popular frameworks include Scrum and Kanban.
8. **Functional Requirements**: Specifications that describe what a system should do, focusing on it’s features and tasks. Examples include system behaviors like login functionality or report generation. These are critical for ensuring the system meets user needs.
9. **Non-functional Requirements**: Focuses on how a system performs rather than what it does. It includes performance, scalability, security and usability. Non-functional requirements ensure a system’s quality and user satisfaction.
10. **MOSCOW Technique**: A prioritization framework categorizing requirements into Must-have, Should-have, Could-have, Won’t-have. It ensures critical requirements are delivered first while managing stakeholder expectations effectively.
11. **FURPS Model**: A framework for categorizing requirements into Functionality, Usability, Reliability, Performance and Supportability. It ensures a balance focus on technical and non-technical aspects of a project.
12. **User Stories**: Short, simple descriptions of a feature or requirement from the end-user’s perspective. Written in the format “As a [user/role], I want to [goal], so that [benefit]” user stories ensure the focus remains on delivering the value.
13. **Feasibility Study**: A preliminary analysis to determine the practicality of project. It evaluates technical, financial, and operational factors to assess whether the project is viable and worth pursuing.
14. **Change Management**: The process of managing changes in project scope, systems, or processes to minimize disruption. It involves assessing impacts, communicating changes and ensuring stakeholder alignment.
15. **Key Performance Indicator (KPI)**: Quantifiable metrics used to measure the success of a project, process or business initiative. Examples include customer satisfaction score, revenue growth, or process efficiency improvements.
16. **Personas**: Fictional representation of user types based on research. Personas help understand user needs, goals, and behaviors, ensuing system meets end-user expectations.
17. **Lesson Learned**: A review conducted after a project to document successes, challenges, and areas for improvement. This knowledge helps future projects avoid similar issues and replicate successful practices.
18. **Traceability Matrix**: A document that links requirements to their origin and tracks them throughout the project lifecycle. It ensures that all requirements are addressed and changes are managed effectively.
19. **RACI Matrix**: A framework that defines roles and responsibilities in a project. RACI stands for Responsible, Accountable, Consulted, and Informed, ensuring clarity and accountability among team members.
20. **Data Flow diagrams (DFDs)**: Visual tools that show how data moves through a system, from input to processing to output. DFDs helps understanding and documenting system functionality and identifying bottlenecks or inefficiencies.