FORUMS

1. **Requirements**

A **requirement** is a specific need or condition that a system, product, or project must fulfill to achieve its goals. It can be functional (describing what the system should do) or non-functional (addressing quality attributes like performance, security, or usability). Requirements are typically documented and serve as a foundation for design, development, and validation.

1. **Stakeholder**

A **stakeholder** is any individual, group, or organization that is interested in or is affected by the outcome of a project, system, or product. Stakeholders can include customers, users, project sponsors, team members, and vendors. They may provide requirements, feedback, and resources, and influence decisions throughout the project lifecycle.

1. **Timesheets**

**Timesheets** are tools used to track and record the amount of time employees spend on specific tasks, projects, or activities over a given period. They can be in physical or digital formats and are commonly used for payroll, project management, or performance evaluation.

1. **Scope Creep**

**Scope creep** refers to the gradual expansion or uncontrolled changes in a project's scope without corresponding adjustments to time, budget, or resources. It often occurs when new features, tasks, or requirements are added after the project has started, often without proper approval or documentation.

1. **Fixed bid projects**

**Fixed-bid projects** are agreements where the project scope, deliverables, and price are predetermined and agreed upon before the work begins. In this model, the client pays a fixed amount regardless of the time or resources required to complete the project.

1. **Billing projects**

**Billing projects** are those projects where payments are based on the work performed, often tied to time spent, resources used, or specific milestones achieved. Common billing methods include **Time and Material,** where charges are based on actual hours and materials, and **Milestone-Based Billing**, where payments are made upon completing predefined deliverables.

1. **Risk**

**Risk** is the potential for an event or condition to occur that may negatively affect a project's objectives, such as scope, timeline, cost, or quality. Risks can arise from uncertainties, external factors, or internal challenges, and they can impact a project's success. Effective risk management involves identifying, analyzing, and mitigating risks to minimize their impact.

1. **Risk analysis**

**Risk analysis** is the process of identifying, assessing, and prioritizing potential risks that could affect a project or objective. It involves evaluating the likelihood of each risk occurring and its potential impact on the project's scope, timeline, budget, or quality. The goal is to develop strategies to mitigate, transfer, accept, or avoid these risks to ensure project success.

1. **Functional requirements**

**Functional requirements** define the specific behaviors, features, or functions a system or application must perform to meet user needs. They describe what the system should do, such as processing data, generating reports, or enabling user interactions. These requirements are essential for designing and developing the system and are typically documented in detail for clear communication among stakeholders.

1. **Non-functional requirements**

**Non-functional requirements** specify the qualities or attributes that a system must have, such as performance, security, reliability, scalability, and usability. Unlike functional requirements, which define what the system should do, non-functional requirements focus on how well the system performs its functions. These requirements ensure the system meets specific standards for efficiency and user experience.

1. **Wireframes**

**Wireframes** are low-fidelity visual representations of a website or application's layout and structure. They outline the placement of elements like navigation, buttons, images, and content, focusing on functionality and user flow rather than design details. Wireframes serve as blueprints for designers and developers, helping to clarify the user experience before detailed design and development begin.

1. **Test case document**

A **test case document** is a detailed document that outlines the conditions, inputs, actions, expected results, and pass/fail criteria for testing a system or application. It specifies how a particular feature or functionality should be tested to ensure it works as intended. Test case documents help testers perform systematic checks.

1. **Database schema**

A **database schema** is a structure that defines how data is organized and how relationships between different data elements are managed within a database. It includes the tables, fields, data types, indexes, and relationships between tables. The schema serves as a blueprint for the database and helps ensure data consistency and integrity.

1. **Change request**

A **change request** is a formal proposal to modify a project’s scope, deliverables, timeline, or other aspects after the project has started. It is typically documented and reviewed to assess its impact on the project before approval or rejection. Change requests help ensure that any modifications are controlled and aligned with the project's objectives.

1. **UAT**

**User Acceptance Testing** is the final phase of testing in which end users or clients validate that a system or application meets their requirements and is ready for deployment. It focuses on real-world scenarios to ensure the system functions as expected in actual use. Successful UAT confirms that the product is acceptable and fulfills the agreed-upon specifications.

1. **SWOT analysis**

**SWOT analysis** is a strategic planning tool used to evaluate an organization, project, or individual's **Strengths**, **Weaknesses**, **Opportunities**, and **Threats**. Strengths and weaknesses are internal factors, while opportunities and threats are external. This analysis helps identify areas for growth, mitigate risks, and make informed decisions to achieve goals effectively.

1. **Elicitation techniques**

**Elicitation techniques** are methods used to gather requirements, insights, and information from stakeholders during a project. Common techniques include interviews, workshops, surveys, brainstorming, document analysis, and prototyping. These techniques help business analysts understand stakeholder needs, clarify objectives, and define project requirements accurately.

1. **MVC Architecture**

MVC Architecture(Model-View-Controller) is a design pattern that separates an application into three interconnected components

* **Model**: Manages data, logic, and business rules.
* **View**: Displays data and handles the user interface.
* **Controller**: Acts as an intermediary, processing user input and updating the Model and View.  
  This separation improves code modularity, maintainability, and scalability.

1. **Conflict management**

**Conflict management** is the process of identifying, addressing, and resolving disagreements or disputes in a constructive manner. It involves using strategies like negotiation, collaboration, and compromise to minimize negative impacts while fostering understanding and teamwork. Effective conflict management ensures a productive and harmonious working environment.

1. **Impediments log**

An **Impediments Log** is a document or tool used in Agile projects to track and manage obstacles that hinder the progress of a team. It lists issues such as resource shortages, technical challenges, or external dependencies and records their status and resolution. Maintaining an impediments log helps teams address blockers efficiently and ensure smooth project progress.