**1) Types of Requirements**

Requirements are categorized into:

* Business Requirements: High-level goals of the organization.
* Stakeholder Requirements: Needs of stakeholders.
* Functional Requirements: Specific functionalities the system must provide.
* Non-functional Requirements: System performance, security, scalability, etc.

**2) What is Business Process Modeling?**

A Business process has goals, specific inputs, specific outputs, uses resources, has a number of activities perform in some order, creates value of some kind for the customer (internal or external).

A business process is a collection of activities designed to produce a specific output for a particular customer or market.

It helps stakeholders understand and analyze workflows, identify inefficiencies, and optimize processes for better outcomes.

**3) What is Scope Creep?**

Scope creep is when a project's requirements, deliverables, budget, timeline, or resources change without approval after the project has started.

This can lead to: Increased costs, Longer timelines, Missed deadlines, and Client dissatisfaction.

Effective documentation and stakeholder communication help manage scope creep.

**4) What is 3-Tier Architecture?**

3-Tier Architecture divides an application into three layers:

* **Presentation Layer**: User interface.
* **Application Layer**: Business logic.
* **Data Layer**: Database and data management.

This structure ensures scalability, maintainability, and separation of concerns.

**5) How do you handle conflict management? What technique do you use?**

Conflict management is the process of handling disputes and disagreements between two or more parties.

We use Thomas Kilman Technique, where x Axis is Cooperation and Y-Axis is Assertiveness.

5 Options for Conflict Management



**6) What is RUP?**

The Rational Unified Process (RUP) is a software development framework that divides a project into four phases: **Inception, Elaboration, Construction, and Transition**.

It’s iterative and focuses on delivering incremental value.

**7) What is Waterfall Model?**

The Waterfall model is a linear and sequential approach where phases like requirements, design, development, testing, and deployment are completed one at a time.

Each phase must be completed entirely before the next phase begins.

It’s suited for projects with stable and well-defined requirements.

At the end of each phase, a review takes place to determine if the project is on the right path. Whether to or not to proceed with the project or discard it.

Works for the well-defined requirements.

**8) What is V Model?**

The V model (Verification and Validation model) is a software development lifecycle model where every development phase is directly associated with a testing phase, ensuring that each deliverable is verified and validated.

Works well for smaller projects.

**9) What is Spiral Model?**

The Spiral Model is an iterative and risk-driven development approach.

It combines elements of Waterfall and prototyping by cycling through planning, risk analysis, development, and evaluation repeatedly.

This model gives more emphasis on risk analysis.

Software is produced early in the life cycle.

Good for large projects.

**10) What are the Four Main Values of Agile Manifesto?**

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

**11) What is Product Backlog? What is Sprint Backlog?**

* **Product Backlog**: A prioritized list of all requirements and features for the product.
* **Sprint Backlog**: A subset of the product backlog selected for a specific sprint, including tasks to deliver those items.

**12) What is Epic?**

An Epic is a large user story or a feature that is too broad to be completed in a single sprint.

It is broken down into smaller, manageable user stories.

**13) What is User Story?**

A User Story is a short, simple description of a feature from the user’s perspective, typically written as: "As a [role], I want [feature] so that [benefit]."

**14) What is Product Burndown Chart?**

A Product Burndown Chart is a visual representation that shows the remaining work in the product backlog against the project timeline, helping teams track progress and predict project completion.

**15) What is SMART?**

SMART is a framework useful for setting clear, actionable, and achievable objectives, ensuring they are Specific, Measurable, Achievable, Relevant, and Time-bound.

**16) What Techniques do you use to set a Priority of User Stories?**

Techniques like MoSCoW (Must Have, Should Have, Could Have, Won’t Have), Kano Model, and Weighted Shortest Job First (WSJF) are used to prioritize user stories based on business value and urgency.

**17) What is BV? What is CP? What Techniques Do You Use for the Same?**

* **BV (Business Value)**: Measures the value a feature brings to the business. We use currency technique here.
* **CP (Cost of Delay/Complexity Points)**: Quantifies the cost or effort of delay. We use Fibonacci technique.

**18) What is Abstraction? (MVP – minimum Viable product)**

Abstraction is the process of simplifying complex systems by focusing only on essential details while hiding irrelevant information.

It’s commonly used in system design and object-oriented programming.

**19) What is Gold Plating ?**

Gold Plating is a practice of making changes to a project that are outside of the original agreed upon scope.

Gold plating takes time, generally seen as a poor practice.

**20) What is Encapsulation?**

Encapsulation uses information hiding technique to hide the complex internal structure or working.

Encapsulation is a fundamental concept in object-oriented programming (OOP) that involves bundling data (variables) and the methods (functions) that operate on that data into a single unit, typically a class.

It restricts direct access to some of the object's components, ensuring that sensitive data is hidden and protected from unintended interference or misuse. Instead, controlled access is provided through publicly exposed methods, often called getters and setters.

**21) What is Generalization? What is Association?**

* **Generalization**: A relationship where one entity is a specialized version of another (e.g., a "Car" is a generalization of "SUV").
* **Association**: A relationship between two objects that interact with each other but are independent.

**22) What is Aggregation? What is Composition?**

* **Aggregation**: A "has-a" relationship where one object contains another, but they can exist independently (e.g., Library and Books).
* **Composition**: A stronger "has-a" relationship where one object cannot exist without the other (e.g., Human and Heart).

**23) What is Inheritance?**

Inheritance is an object-oriented programming concept where one class (child) derives properties and behavior from another class (parent), enabling code reusability and hierarchy.

**24) What is Polymorphism?**

Single Instructions multiple operations.

Polymorphism allows objects to take on multiple forms.

In programming, it enables a single interface to handle different types of objects, such as method overloading and overriding.