UNIT-1

**Q) Water Fall Model and Agile model**

Ans) Waterfall is a linear framework where each phase is completed before moving to the next. It emphasizes documentation and planning, making it suitable for projects with clear, fixed requirements. However, it lacks flexibility and can be risky if changes are needed later.

Agile Software Development is a set of principles and methodologies aimed at delivering software incrementally and iteratively, emphasizing collaboration, flexibility, and customer satisfaction. It provides an alternative to traditional project management approaches like the Waterfall Model and is particularly effective in dynamic environments where requirements and priorities change frequently.

**Q) Identify effective communication principles in a distributed software development team**

**Ans)** 1. Clear and Concise Communication

2. Regular and Structured Meetings

3. Effective Use of Collaboration Tools

4. Time Zone Awareness and Flexibility

5. Written Documentation and Knowledge Sharing

6. Over-Communication

7. Fostering a Collaborative and Inclusive Culture

8. Video and Audio Communication for Personal Connection

**Q) Identify how the different layers of Software Engineering Layered Technology work together to ensure the quality of the software development process.**

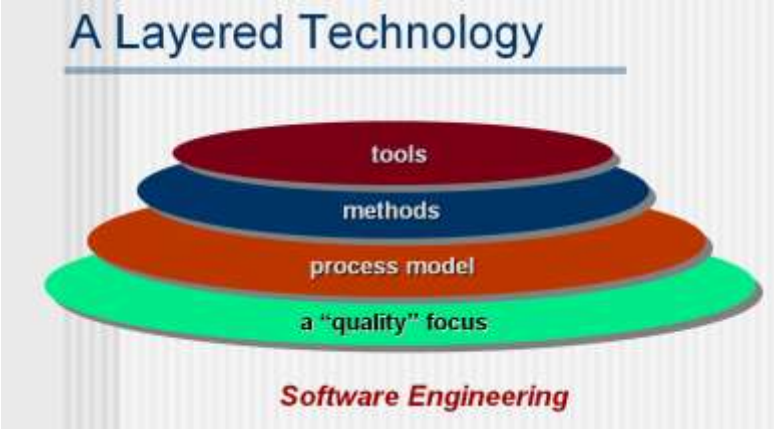
**Ans)** The layers of software engineering include:

• Tools: Support the development process.

• Methods: Provide technical knowledge and processes.

• Process: Defines the workflow and activities.

• Quality Focus: Ensures the software meets quality standards. These layers ensure structured development, resulting in highquality software.




**UNIT-2**

**Q)Identify the Design concepts and discuss them with examples.**

**ANS) Abstraction:** Hides complex details.

**Modularity:** Breaks down the system into independent modules.

**Coupling:** Measures module interdependence (lower is better).

**Cohesion:** Measures module's functional unity (higher is better). Violating these can result in tightly coupled code or poor scalability.

**Q) Examine various architectural styles used in software engineering**

**ANS) 1.** Layered Architecture

**2.** Client-Server Architecture

**3.** Microservices Architecture

**4.** Event-Driven Architecture

**5.** Service-Oriented Architecture (SOA)

**6.** Model-View-Controller (MVC)

**7.** Pipe-and-Filter Architecture

**UNIT-3**

Q) **Identify the need for the Golden rules for user interface design**

ANS) Golden Rules:

**1) Place the User in Control**

Users should feel that they are in charge of the interface and can make decisions without unintended consequences. This involves providing undo options, navigational freedom, and meaningful feedback.

**2) Make the Interface Consistent**

Consistency in layout, colors, terminology, and interaction patterns helps users predict outcomes and navigate easily, reducing their cognitive load.

**3) Offer Error Prevention and Simple Error Handling**

Systems should be designed to **prevent user errors** before they occur.

## Q) Types Of Testing

**Unit Testing:**

Unit testing involves testing individual components or functions of a software application in isolation to ensure that each unit behaves as expected.

**Integration Testing**

Integration testing verifies that multiple components or modules work together correctly when combined.

- Focuses on interactions between modules.
- Conducted after unit testing.

### Black Box Testing

Black Box Testing is a testing method where the internal structure or code of the application is not known to the tester. The focus is on testing functionality based on input and output.

### White Box Testing

White Box Testing (also called Clear Box or Glass Box Testing) involves testing the internal logic

and structure of the code. The tester knows the source code and how the system works internally.

## UNIT-4

**Q) Analyze the key components of an RMMM plan**

**(OR)**

**Apply the RMMM plan strategy to any real time example**

**ANS)** Risk Identification: Recognizing project risks

• Risk Mitigation: Planning actions to reduce risk

• Risk Monitoring: Regularly assessing risks

• Risk Management: Taking actions when risks materialize Example: Budget risk mitigated by scope adjustment

**Q) Analyze how McCall's Triangle of Quality Factors supports a balanced approach to software quality**

**Ans)** Product Operation: Correctness, Efficiency, Reliability

• Product Revision: Maintainability, Testability (e.g., modular design)

• Product Transition: Portability, Reusability (e.g., cloud migration support) All ensure overall product quality