ASSIGNMENT-1.

Q1 Differentiate between "process" and "product" in the context of Software engineer.

Process: A set of activities, methods and practices followed to develop software efficiency. It includes requirement analysis, design, coding, testing, deployment and maintenance

Product: The final software or system that is deliverd to the end-users after following the Software development process.

Example:

- Process → Agile, Waterfall, Spiral, RAD, etc.
- Product → Microsoft Word, Android OS, Web application etc...

Q2 Explain the concept of software engineering as a layered technology and describe each layer briefly.

Software engineering is structured as a layered technology, meaning it is built an multiple layers, each serving a specific purpose in ensuring high-quality software development. These layers provide a systematic approach to software development and maintenance.

Layers of Software Engineering:

1. Quality Focus: Ensures that software meets reliability, maintainability, and performance standards.

2. Process Layer: Provides a framework for planning, managing and executing development tasks.

3. Methods Layer: Encompasses techniques and best practices used in software design, development, testing and deployment.

4. Tools Layer: Supports the implementation of process and methods with automation and efficiency.

Q3 Explain the Rapid Application Development (RAD) Model and its application in time-constrained projects.

RAD Model: A software development model that focuses on rapid prototyping and iterative development. It minimize planning and emphasizes user feedback.

Key Phases:
1. Bussiness Modeling
2. Data Modeling.
3. Process Modeling.
4. Application Generation
5. Testing & Deployment

Application in Time-Constrained Projects:
1. Best suited for projects with tight deadlines.
2. Ideal for applications where user requirements change frequently.
3. Commonly used for UI-heavy applications.

Q4 Explain Scrum as an Agile Development Model for Software Engineering.

Scrum is a popular Agile Framework used in software development that focuses on iterative and incremental progress through short development cycles called sprints. It promotes collaboration, flexibility and continuous improvement to deliver high-quality software efficiently.

Key elements of Scrum:

1. Roles in Scrum:

→ Product Owner: Defines the product vision, manages the product backlog, and ensures the team works on the highest priority tasks.

Development Team: A self-organizing team responsible for delivering the product increment.

2. Scrum Artifacts:

→ Product Backlog: A prioritized list of features, enhancements and bug fixes to be developed.

Sprint Backlog: A list of tasks selected for the current sprint.

3. Sprints:

→ A sprint is a time-boxed development cycle. (typically 1 to 4 weeks)

→ The goal is to deliver a working increment of software at the end of each sprint.

Q5 Explain Spiral Model with a neat diagram

A spiral model is a risk-driven software development model that combines iterative development with elements of the waterfall model. It is designed to handle large, complex and high-risk projects by focusing on continuous risk assessment and refinement

Phases of Spiral Model:

1. Planning:
Identify project objectives, constraints and alternative solutions.

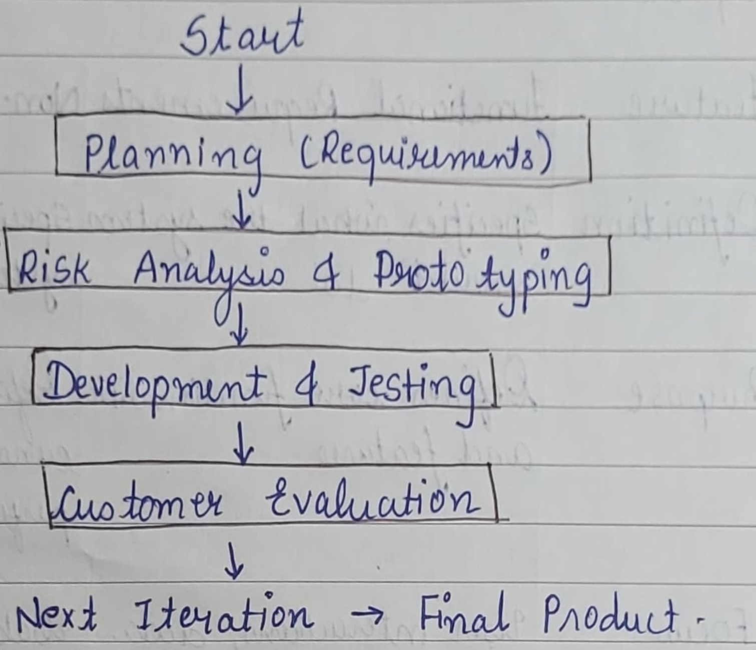
2. Risk analysis:
Identify potential risks and develop strategies to mitigate them

3. Engineering:
Design, develop and test the product incrementally.

4. Evaluation
Changes and refinements are incorporated before proceeding to the next iteration.

→ After each cycle, the project moves towards completion, gradually refining the system until the final product is delivered.

Diagram:



Q6 Define Software Engineering and explain its significance in the development process.

Software engineering:

→ It is the systematic application of engineering principles to software development for ensuring reliability, efficiency and scalability.

Significance:

1. Reduces software complexity.
2. Ensures software quality and maintainability.
3. Promotes cost-effective development.
4. Facilitates collabration among teams.

Q7 Difference between Functional and Non-Functional Requirements.

Feature	Functional Requirements.	Non-Functional Requirement
Definition	Specifies what the system should do.	Specifies how the system should perform
Purpose	Defines core functionalities and features	Defines system properties, constraints and performance.
Focus	User interactions, data processing, and bussiness rules.	Usability, security, reliability, performance, scalability.
Example	User can log in using email and password.	The system should respond within 2 seconds.