



Question 1: [25 marks]

a) Elaborate on each of the five (5) stages of the waterfall model.

The Waterfall model is a sequential design process, often used in software development processes. The five stages include:

1. **Requirements Analysis:** Gathering detailed information about the system's requirements from stakeholders.
2. **System Design:** Designing the system architecture and deciding how the requirements will be implemented.
3. **Implementation (Coding):** Writing the actual code based on the system design.
4. **Integration and Testing:** Integrating various system modules and thoroughly testing the software for defects.
5. **Maintenance:** After deployment, any issues found are fixed, and enhancements may be made as needed.

b) Explain why incremental development is the most effective approach for developing business software systems. Why is this model less appropriate for real-time systems engineering?

Incremental development allows for gradual development and delivery of functional parts of the system, reducing risks and allowing feedback from stakeholders throughout the process. It's effective for business software because requirements often evolve as users provide feedback. However, it's less suited for real-time systems engineering due to the need for strict timing and synchronization, where small incremental changes could disrupt real-time system performance and create inconsistencies.

c) Discuss the possible benefits and problems of using the re-use software model in a software development project.

Benefits:

- Reduced development time and cost.
- Increased reliability as reused components are often well-tested.
- Lowered risk as the reused software has already been tested.

Problems:

- Compatibility issues with new systems.
 - The reused code may be outdated or may not fit all new requirements.
 - Hidden bugs or security vulnerabilities may still exist in reused components.
-

Question 2: [25 marks]

a) Explain what use cases and use case diagrams are, and how they can be used to capture the functional requirements for a software system.

A **use case** is a description of how users interact with a system to achieve a specific goal. It outlines the steps involved in a process and the user's objectives. **Use case diagrams** visually represent these interactions between users (actors) and the system, helping capture functional requirements. They help stakeholders and developers understand what the system should do without specifying how it does it.

b) Draw a use case diagram for a simple ATM machine system.

A simple use case diagram for an ATM machine system would include actors such as "Customer" and use cases like:

- Withdraw Money
- Check Balance
- Deposit Money
- Transfer Funds

Actors and use cases are connected by lines to represent interactions.

Question 3: [25 marks]

a) Explain the importance of using tools for source code management.

Source code management (SCM) tools, like Git, allow developers to track changes, collaborate with others, and manage versions of the codebase. They ensure that previous versions can be restored if necessary and reduce conflicts when multiple developers work on the same project.

b) Outline the code management best practices.

1. Commit frequently and in small increments.
2. Write clear, meaningful commit messages.
3. Use branches for feature development and testing.
4. Regularly merge branches to keep the codebase up-to-date.
5. Always review and test code before merging.

c) Testing is an important aspect of software development and maintenance. Elaborate on the purpose of each of the following different types of testing:

- **i) Unit testing:** Focuses on testing individual components or units of the software to ensure they function correctly.
- **ii) White Box testing:** Involves testing the internal structure of the code, verifying logic, and the flow of the software.
- **iii) Black Box testing:** Tests the software without knowledge of its internal workings. It focuses on inputs and expected outputs.
- **iv) Integration testing:** Ensures that different modules or components of the system work together correctly.

d) Discuss the difference between functional and non-functional requirements in software engineering.

Functional requirements specify what the system should do, such as processes, data handling, and interactions.

Examples: User authentication, transaction processing.

Non-functional requirements define the system's quality attributes, like performance, security, and usability.

Examples: The system should respond within 2 seconds, and the system must handle up to 10,000 users concurrently.

Question 4: [25 marks]

a) Outline any six (6) principles of the Agile Manifesto.

1. Customer satisfaction through early and continuous software delivery.
2. Embrace changing requirements, even late in development.
3. Deliver working software frequently (in weeks rather than months).
4. Close, daily cooperation between businesspeople and developers.
5. Build projects around motivated individuals.
6. Face-to-face conversation is the best form of communication.

b) Differentiate between scrum and a sprint.

- **Scrum** is an Agile framework used to manage software development, focusing on iterative progress through "sprints."
- A **sprint** is a time-boxed period (usually 1–4 weeks) in which a specific set of features or functionalities is developed.

c) Justify the importance of a sprint review.

A sprint review is critical as it allows the team to showcase what has been developed during the sprint. Stakeholders provide feedback, ensuring that the project remains aligned with business goals and adjusting the next sprint's focus if needed.

d) Explain what you understand by Digital Transformation and why it is important to go for Digital Transformation since the COVID-19 pandemic.

Digital Transformation refers to integrating digital technologies into business processes, fundamentally changing how organizations operate and deliver value to customers. Since COVID-19, digital transformation became crucial due to:

- Increased reliance on remote work.
 - The need for businesses to adopt online platforms for continuity.
 - Enhanced customer expectations for digital services and communication.
-