

a) Relevance of Feasibility Study in Today's Software Development Economy (3 marks)

A **feasibility study** evaluates whether a proposed software project is viable before significant resources are committed. Its relevance includes:

- **Risk Reduction:** Helps identify technical, financial, and operational risks early.
- **Cost-Benefit Analysis:** Ensures that the project is economically justified.
- **Informed Decision-Making:** Provides stakeholders with evidence to support project approval or rejection.

b) Rapid Application Development (RAD) Methodology (6 marks)

Rapid Application Development (RAD) is a software development methodology that emphasizes quick development and iteration through:

- **Prototyping:** Continuous delivery of working prototypes for user feedback.
- **User Involvement:** Frequent user engagement ensures that the system meets real needs.
- **Component-Based Development:** Uses reusable components to accelerate development.
- **Short Development Cycles:** Focuses on speed and flexibility over rigid planning.

Advantages:

- Faster delivery of software
- High user satisfaction due to involvement
- Reduced development risk

c) Difference Between User Requirements and System Requirements (4 marks)

- **User Requirements:** High-level statements describing what the user needs the system to do.
Example: “The system should allow users to generate sales reports.”
- **System Requirements:** Detailed descriptions of what the system must do to fulfill user requirements.
Example: “The system shall generate monthly sales reports in PDF format including total sales per product.”

Key Difference: User requirements are conceptual and focus on goals, while system requirements are technical and implementation-specific.

d) Functional and Non-Functional Requirements (6 marks)

Functional Requirements define specific behavior or functions of the system:

- *Example 1:* The system shall allow users to log in with a username and password.
- *Example 2:* The application shall generate an invoice after a purchase.

Non-Functional Requirements specify system qualities or constraints:

- *Example 1:* The system should load the dashboard within 3 seconds.
- *Example 2:* The software must comply with GDPR data protection standards.

e) Four Factors Affecting Success/Failure of IT Systems (8 marks)

1. **Lack of Security:**

Systems without proper security measures are vulnerable to breaches, leading to data loss or legal issues.

2. **Poor Planning:**

Inadequate planning results in scope creep, missed deadlines, and budget overruns.

3. **Poor Communication:**

Misunderstandings between developers, users, and stakeholders can lead to systems that do not meet user needs.

4. **Lack of Technical Knowledge:**

Teams lacking the required skills may produce low-quality, inefficient, or non-functional systems.

f) Role of External Requirements in Software Development (3 marks)

External requirements are constraints imposed by external bodies such as legal, regulatory, or contractual rules.

- They **affect software quality** by ensuring compliance with standards (e.g., accessibility, security).
- Failure to meet these can lead to **penalties, lawsuits, or rejection** of the system.

g) Define: Sprint and Prototype (4 marks)

- **Sprint:** A short, time-boxed period (usually 1–4 weeks) in agile methodologies during which a specific set of features is developed, tested, and potentially released.
Used in: Scrum.
- **Prototype:** A simplified version of the system or component built to demonstrate functionality and gather feedback before full development.
Used in: RAD and iterative development.