

# Introduction to Software Engineering

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

## Module:

### ~~Definition of Software Types & Characteristics of Software~~

#### Types of Software

- System Software
- Application Software
- Embedded Software
- Engineering/Scientific Software
- Web & Mobile Apps.

Operational Characteristics  
These related to how well the software performs during actual use.

#### Functionalability

#### Reliability

Efficiency : Characteristic

associated to how easily the software can be moved or adapted to different environments.

Portability : The ability to run on different hardware or operating systems with minimal changes or Example: A web application that works on Windows, Mac and Linux.

Smoothly : Reusability (the software like modules or parts of libraries) should be usable in other projects.  
Example : A login authentication module being reused in multiple web apps.

Maintainability Characteristics  
related to how easily the software can be modified or extended after deployment.

•

Modularity: The Software should be divided into independent modules so that changes in one don't affect others's Ex - ~~Don~~ In an e-commerce app product listing and payment gateway can be separate modules.

Scalability: The Software should handle increasing work loads (more users more data) without performance degradation.

- Software Attributes - The abilities:
- Functional Correctness
- Usability & Maintainability
- Efficiency & Performance
- Scalability & Adaptability.

What is Software Engineering?

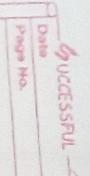
- Ray K Malli: Engineering discipline.
- For reliable software
- IEEE: Systematic, disciplined approach
- Layers: Process → Methods → Tools.

Why Software Engineering?

- Manage complexity.
- Improve quality
- Control costs & schedules
- Handle changing requirements.

Software Cost > Hardware Cost

Hardware is a one-time physical investment (e.g. buying a server) - Once purchased its cost is mostly fixed.



- Software, on the other hand, involves:
  - Development Cost  $\rightarrow$  designing, coding, testing, documentation
  - Maintenance Cost  $\rightarrow$  fixing bugs, updating features, adapting to new environments

- Key Cost Drivers in Software Engineering:
  - Complexity: More complex systems require more effort to design, test, and maintain.
  - Maintenance: Software changes frequently (bug fixes, security updates, new features). Studies show 70-80% of total cost is spent on deployment.
  - Changing Requirements
  - Business needs evolve  $\rightarrow$  requires rework and adaption and increasing cost.

How to Estimate Software Costs?  
Models like COCOMO (Constructive Cost Model) are used.

- They consider:
  - Size of Software (measured in KLOC = thousand lines of code)
  - Complexity Level (Simple, intermediate, embedded)
  - Team experience & tools
  - Output  $\rightarrow$  Effort (person-month)

### Managing Costs

- Use modular design
- Promote reuse
- Employ proper process & CASE tools

Q1) Which of the following best describes software?

Q1.) What is Software?

A set of instruction given to the computer.

Ans: Q2.) Which of the following is not a type of software

Ans: Cable

Q3.) What are the main characteristics of good Software?

Ans: Efficiency.

Q4.) Which form defines the process of developing software in a systematic way?

Ans: Software Engineering.

Q5.) Which of the following is an attribute of good software?

Ans: Maintainability.

Q6.) What is a key challenge faced by software engineers?

Ans: Coping with legacy system

Q7.) What type of Software supports the running of Application Software?

Ans: System Software.

Q8.) What is Software cost usually associated with?

Ans: Development and Maintenance.

Q9.) Who is known as the father of Software Engineering?

Ans: Barry Boehm.

Q10.) Software that controls and manages the hardware components is called?

Ans: System Software

Q11.) Which Software attribute refers to how well the Software works under stated conditions?

Ans: Reliability.

Q12)

Which of the following is a goal of Software engineering?

Ans: Producing reliable and efficient software.

Q13) Which of these is a key activity in software engineering?

Ans: Requirement Analysis.

Q14) Software with minimal human interaction for its functions is called?

Ans: Automated Software.

Q15) Which is a software attribute related to time and resource usage?

Ans: Efficiency.

Q16) Identifying the reason Software engineering is considered essential in modern development practices?

Ans:

To Reduce software cost and improved quality

Q17) Recognize the challenge in Software engineering that is specifically related to scalability issues?

Ans: Handling large and growing system.

Q18) Infer the significance of system engineering knowledge in the software development life cycle.

Ans: It integrates software with other system components.

Q19) Identify how documentation contributes to the maintainability of a software system.

Ans: ~~It helps~~ It helps maintenance and future development

Q20) Indicate the benefit of modularity in improving software design.

Ans: It affects system performance.

Q21) Recognize which attribute of software contributes to its continued current operation over time.

Ans: Reliability.

~~UN~~ SUCCESSFUL

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

Q22) Infer why accurate cost estimation remains a persistent challenge in software engineering.

Ans: Many variables and unpredictable projects.

Ans: Many variables and unpredictable projects.

Ans: Review the components typically changes.

Ans: Development and maintenance includes cost increase engineering cost estimation includes.

Ans: Development and maintenance cost.

Ans: Interpret the importance of capturing user requirement during software development.

Ans: To develop appropriate and useable software.

Q23) Indicate how abstraction simplifies the complexity of software design.

This hides complexity.

Q24) Define Software Engineering.

Ans: Any attributes of good software.

Q25) State two types of software with example.

Q26) Identify the key challenges faced by software engineering engineers.

Q27) Recall the importance of system engineering in software development.

Q28) Explain the characteristic of software.

Q29) Discuss how software engineering helps in reducing cost.

Q30) Illustrate the difference between software and hardware.

Q31) Classify different types of software engineering cost.

Q32) Summarize the benefits of using a systematic software process.

Q33) Comparative study between iterative process and incremental delivery?

Q34) Challenges of incremental delivery?

Q26) Identify how software differs from hardware in terms of functionality and behaviour?

Ans: Software is logical and intangible.

Q27) Identify the main focus of system engineering in the context of Software development.

Ans: Integration of Hardware and Software.

Q28) Interpret what the key challenges in Software engineering typically refers to.

Ans: Managing complexity, cost and quality.

Q29) Inter why understanding the characteristic of Software is important for developers?

Ans: To choose the best development approach.

Q30) Recognize why System Software are

considered engineered rather than just

coded.

Ans: They follow systematic development process.

Module - 2

[3] marks

Q1) Explain the concept of Software process model.

Q2) Summarize the characteristic of Waterfall model?

Q3) Discuss the purpose of process iteration.

Q4) Identify the benefits of incremental delivery?

Q5) Apply RAD model to a time constraint project?

Q6) Discuss the properties of RAD model?

Q7) Illustrate the benefits of Agile model over Waterfall model

Q8) Demonstrate components based development in real application, Use reusable module like API or plugin.

Q9) Use prototyping to gather early feedback (Hint: Iterative model)

Q10) Choose an appropriate model for unstable requirement (Hint: Spiral or Hizd model):

Requirement

## M<sup>2</sup>-5 Marks

Page No. \_\_\_\_\_

- 1.) Explain the features and drawbacks of Waterfall model?
  - 2.) Compare Iterative and incremental development models?
  - 3.) Apply the spiral model in high risk planning. (Hint: Focus on high risk project refinement in each cycle).
    - (i) Demonstrate the use of agile methods in real life project development. (Hint: go through modules continuous feedback and focus on risk analysis and iterative learning)
    - (ii) Exploratory development mainly means refining the same system with feedback.
- Q2.) Which model is rigid but simple to understand
- Q3.) What is the main goal of the waterfall prototyping?
- Ans:- Clearly requirements when discard prototype management, which model is complex but best for risk Spiral
- Ans:- Which is a drawback of the incremental model,
- Ans:- Needs proper modular design.
- b.) Which model allows customers to use partial software early
- Ans:- Incremental
- f.) Which model is best for unclear and changing requirements?
- Ans:- Evolutionary
- g.) Which model overlaps risk analysis and development?
- Ans:- Spiral
- g.) Which model combines iterative development

With risk analysis

Ans:

Opportunity

Ans:

What is the main benefit of the iterative model?

Ans:

What is the main benefit of the evolutionary development approach?

Ans:

Which approach of evolutionary development discards the prototype after clarifying requirements?

Ans:

Which approach of evolutionary development follows the same system repeatedly?

Ans:

Exploratory development.

Ans:

What is the main focus of evolutionary development?

Ans:

Gradual refinement with feedback.

Ans:

When do customers receive the full software in the waterfall model?

Ans:

After the final phase

Ans:

Which is a drawback of the waterfall model?

Ans:

Not flexible to changes.

Ans:

Which is an advantage of the waterfall model?

Ans:

Simple and structured

Ans:

When can the next phase start in the waterfall model?

Ans:

After the previous phase is completed

Ans:

What is the first phase in the waterfall model?

Ans:

Requirements

Ans:

Which model is also called a linear sequential model?

Ans:

Waterfall

## Module 2 M.C.Q

Q.1) Interpret the primary purpose of the waterfall model?

Q.1)

Ans:- To follow a sequential development process, identify a core characteristic that defines

Ans:- Q.2)

Ans:- Identify a core characteristic that defines Agile method.

Ans:- Q.3)

Ans:- Iterative development with customer feedback, recognize the concept of process models, iteration within software process models?

Ans:- Q.4)

Ans:- Repeating phrases to improve the product.

Ans:- Q.5)

Ans:- Classify the model that integrates design and prototyping through multiple stages?

Ans:- Q.6)

Ans:- Interpret the main goal of RAD methodology

Ans:- Q.7)

Ans:- To quickly develop prototypes and gather user feedback.

Ans:- Q.8)

Ans:- Interpret the benefit of using component based software engineering in modern projects?

Ans:- Q.9)

Ans:- Reduce cost and improves maintainability, recognize how case tools contribute to

Ans:- Q.10)

Ans:- Different phases of software development?

Ans:- Q.11)

Ans:- Supporting software process activities

Ans:- Q.12)

Ans:- Compare Waterfall and Agile models with respect to flexibility?

Ans:- Q.13)

Ans:- Agile is more flexible than waterfall.

Ans:- Q.14)

Ans:- Identify the primary objective of extreme programming practices?

Ans:- Q.15)

Ans:- To improve software quality through frequent releases

Ans:- Q.16)

Ans:- What is the idea behind incremental delivery in iterative development models?

Ans:- Q.17)

Ans:- Breaking systems into functional increments for delivery

Ans:- Q.18)

Ans:- Interpret the primary purpose of the waterfall model in sequential process management

Ans:- Q.19)

Ans:- Follow a sequential development process.

for Q12) Identify a distinguish characteristics of agile method in handling requirements.

Ans: Creative development with customer feedback.

Q13) Recognize how iteration plays a role in defining & refining during development?

Ans: Repeating Phases to improve the product.

Q14) Classify the hybrid model that merges design with prototyping stages?

Ans:

Q15) Spiral model?

Ans: To support how R&D emphasize with development and frequent user feedback.

Q16) Infer the benefit of component based software engineering in modular development?

Ans: Reduces cost and improves maintainability.

Q17) Illustrated how CASE tools support various phases of software development?

Ans: Supporting Software process activities.

Q18) Compare the flexibility between the Waterfall and agile methodology.

Ans: To improve software quality through frequent releases.

Q19) Define the concept of incremental delivery within iterative model?

Ans: Breaking System into functional increments.

at Q20) Identify the primary goal of extreme programming?

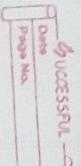
Ans: To improve software quality through frequent release.

Q21) ~~Prototyp~~ Identify the agile principle that supports adapting to changing customer requirements,

Ans: Welcome changing requirements and choose the suitable process model.

Q22)

## Software Prototyping



- Understands requirements better.

- Get early feedback from users.

- Explore technical feasibility.

- The often treated as its own model (like waterfall, spiral, or incremental), but it can also be a part of other models such as the SPIRAL Model & Rapid Application Development.

- Key Properties: Early representation: Shows a working model of the product before full development.

- User Involvement: User review prototypes and provide feedback.

- Iterative Refinement: The prototype is modified until requirements are clear.

- Advantages:

- Helps clarify ambiguous requirements.

- Reduces misunderstandings between users and developers.

- Identifies missing or conflicting features early.

- Improves system usability through feedback loops.

- Disadvantages:

- Cost lead to scope creep if users keep requesting changes.

- May result in poorly designed architecture if the prototype is pushed.

- The prototype is pushed extra cost and time if the prototype is not used in the final system.

- Overview of CAST: CAST refers to the use of software tools

- to support software development and maintenance activities.

- Provides automated support for activities like analysis, design, coding and testing.

- Improves productivity, quality, and

- cost-effectiveness.

consistency across software projects.

• Repetition of repetitive task in software development.

• Graphical models for system design (e.g., UML diagrams).

• Centralized repository for documentation and models.

• Support multiple stages of SDLC.

• Promotes standardization and reuse of components.

Advantages of CASE

1. Increases productivity by automating manual tasks.

2. Improves accuracy and reduces human error.

3. Encourages reuse of software components.

4. Enhances collaboration among team members.

5. Provides consistent documentation and design standards.

Disadvantages of CASE

1. High cost of CASE tools and training requirements.

2. Complexity in integrating CASE tools with existing systems.

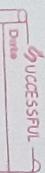
3. Steep learning curve for developers unfamiliar with tools.

4. May lead to over-dependence on tools.

5. Not always flexible for small or agile projects.

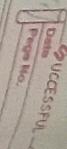
## Module-4

### SAQ 3 Marks



- 1.) Demonstrate unique testing on a logging module?
- 2.) Apply blackbox testing for an E-commerce Cost?
- 3.) Use equivalence partitioning to test input?
- 4.) Illustrate how validation testing is performed
- 5.) Apply Regression testing after bug fixing.
- 6.) Analyze the difference between verification & validation.
- 7.) Distinguish between Alpha & beta testing.
- 8.) Examine how quality standards effect testing.
- 9.) ~~Analyze the relationship between~~
- 5 Marks
- 1.) Analyse the relationship between cohesion and coupling.
- 2.) Compare abstraction and information hiding as complexity reduction technique.
- 3.) Differentiate between architectural patterns and designed patterns based from their scope
- 4.) Examine how modularity contributes to achieving functional dependencies in software systems
- 5.) Classify 7 types of cohesion based on their strength.
- 6.) Analyse the impact of ~~tight~~ coupling on software maintainability
- 7.) Investigate the role of refinement in top down design methodology
- 8.) Compare command line interfaces with graphical user interfaces in terms of usability.
- 9.) Categorize modern GUI elements based on their primary interaction functions
- 10.) Examine the design consideration for effective error message implementation

- 11.) Distinguish between procedural ~~program~~ and object oriented orientation for programming and object analyse the criteria for selecting paradigm.
- 12.) Analyse the criteria for selecting paradigm.
- 13.) Breakdown the key components for specifying application programming languages for specific projects.
- 14.) Compare different coding practices of their benefits for team development interaction.
- 15.) Investigate the relationship between design principal and GUI control selection.
- M-5 3 Marks
- Q1.) Explain how ISO 9000 standard contribute to software quality improvement?
- Q2.) Describe the five levels of the capability maturity model and their importance in software process improvement?
- Q3.) Explain the need for proper software project management to avoid project failure?
- Q4.) Explain how the size of software is estimated during project planning?
- Q5.) Summarise the steps involved in risk mitigation, monitoring and management plan.
- Q6.) Explain how earned value analysis helps track the progress and performance of a software project?
- Q7.) Describe the components of a network scheduling diagram and its role in project planning.
- Q8.) Differentiate between function point analysis and lines of code based estimation methods.
- Q9.) Explain the difference between verification and validation in software testing. Give suitable examples.
- Q10.) Describe the significance of quality assurance in the software developing.



Q.)  
Ans:-

What is Software Design. Process of defining Software Design is the process of defining software components, interfaces and the architecture to satisfy specified requirements.

Q.) Objectives of Good Design.

Correctness and efficiency.

Maintainability and reusability.

Flexibility and scalability.

Clean Structure and readability.

Principles of Software Design

3.) Principles of Software Design

- Divide and Conquer.

- Modularity.

- Abstraction

- Stepwise refinement

- Separation of concerns.

- Concept of Abstraction.

Abstraction focuses on hiding unnecessary details and showing only essential features.

It simplifies complexity and improves clarity.

5.) Types of Abstraction:

Procedural Abstraction: Decompose tasks

into smaller procedures

Data Abstraction: Represent data using abstract data types.

Control Abstraction: Hide control flow

details like loops or calls.

Advantage of Abstraction

Simplifies system understanding

Enhances reusability.

Supports easier maintenance

Reduces complexity.

Q.) What is Software Architecture?

Ans:- Architecture defines the overall structure

of the system, showing components, their

interactions and how they interact with each other.

Q.) What is Software Architecture?

Ans:- Architecture defines the overall structure of the system, showing components, their

interactions and how they interact with each other.

OPPO A17 · © ❤️ suji ❤️ ✨

interactions, and relationships.

8.) Architectural Styles  
Common Architectural Styles

- Layered Architecture.
- Client - Server.

- MVC, Model - View - Controller
- Microservices.

9.) Importance of Software Architecture

- Ensures scalability and performance.
- Improves reusability.
- Simplifies maintenance.
- Provides a foundation for design decisions.

10.) Design a gantt chart for a software project schedule.

11.) Formulate a risk management plan for a e-learning platform?

12.) Describe the purpose of regression testing when it is applied in a project.

13.) Develop a project schedule for a mobile app project.

M5 - Marks

1.) Design a comprehensive testing framework for

software quality assurance, using verification & validation principles. Hint: Testing framework integrates

to ensure correctness, during development with validation techniques like reviews, inspection & static analysis.

System acceptance, testing to conform system needs, user requirements.

2.) Develop a complete project plan for a

restaurant POS system. (include scope, cost estimate, schedule, risk plan); Hint:

3.) Create a cost estimation model using Randic point analysis technique.

4.) Construct a comprehensive risk management

Framework using RMM methodlogy for software projects.

⑤ Generate a complete best case methodology for achieving maximum defect detection coverage.

⑥ Synthesis a project scheduling approach using network scheduling techniques for software development.

⑦ Build a details CRM implementation plan for organisational process maturity improvement.

⑧ Compose an earn value analysis system for real time project performance measurement.

⑨ Plan and implemented people management strategy for software project team effectiveness.