November 21, 2025 • • 4 minute read

Software Engineering Important Questions

Resource provide important questions for exam

Important Questions for Software Engineering

End Semester Preparation

UNIT 1: Introduction to Software Engineering

1. Define Software Engineering.
2. What are the characteristics of good software?
3. Explain the layered technology of software engineering.
4. Differentiate between software and hardware.
5. What are software components? Give examples.
6. Write a short note on the software crisis.
7. Explain software applications in different domains.
8. What are the major challenges in software engineering?
9. Explain the Waterfall Model with its advantages and disadvantages.
10. What is the Incremental Model?
11. Describe the Prototype Model with a neat diagram.
12. Explain the Spiral Model and its advantages.
13. What is the Concurrent Development Model?
14. Define Agile Development.
15. Explain the Agile Process Model.
16. What are the principles of agility?
17. Explain Extreme Programming (XP).
18. List other Agile models and tools used in industries.
19. What are the benefits and drawbacks of Agile models?
20. Compare traditional process models with Agile models.

UNIT 2: Software Project Management

1. What is Software Project Management?
2. What are the main activities involved in software project management?
3. Describe the management spectrum.
4. Explain the relationship between People, Product, Process, and Project.
5. What is the W5HH Principle? Explain each question.
6. Why is team management important in software projects?
7. What are the roles and responsibilities of a project manager?
8. Explain project scope and feasibility analysis.
9. What are the steps involved in effort estimation?
10. Define project scheduling.
11. Explain different types of project scheduling methods.
12. What is staffing in project management?
13. Explain quality planning in software projects.
14. What is risk management?
15. Explain the process of risk identification.
16. Describe risk assessment and control techniques.
17. What is a project monitoring plan?
18. Explain detailed scheduling with examples.
19. Define key performance indicators (KPIs) in project monitoring.
20. What tools are used for project monitoring and control?

UNIT 3: Requirements Engineering

1. What is Requirements Engineering?
2. Explain the steps of the Requirements Engineering process.
3. What is problem recognition?
4. What are the tasks of Requirements Engineering?
5. Define requirement specification.
6. What are functional and non functional requirements?
7. Explain requirement elicitation with techniques.
8. What are the categories of customer requirements?
9. What is a Use Case? Give an example.
10. Explain Functional Specification.
11. What is Requirement Validation?

12. Explain Requirement Analysis with its importance.
13. What is the difference between feasibility study and requirement analysis?
14. Write short notes on requirement documentation.
15. What are the qualities of a good SRS document?

UNIT 4: Structured System Design

1. What are the key design concepts in software engineering?
2. Define software architecture.
3. What are architectural styles and patterns?
4. Explain data design and procedural design.
5. What is object oriented design (OOD)?
6. Explain data oriented analysis with examples.
7. Draw and explain an Entity Relationship (ER) diagram.
8. What is a Data Flow Diagram (DFD)?
9. Define Control Flow Model and explain its significance.
10. What is a Data Dictionary?
11. What is Component Level Design?
12. Explain modularity and abstraction in system design.
13. Compare object oriented and functional oriented design approaches.

UNIT 5: Coding and Unit Testing

1. What are programming principles and guidelines?
2. Explain the concept of structured programming.
3. What are coding standards? Why are they important?
4. Explain incremental development of code.
5. How is code evaluation and review managed?
6. Define unit testing.
7. Differentiate between procedural unit testing and class testing.
8. What is code inspection?
9. Explain Cyclomatic Complexity with an example.
10. What is Halstead Software Science measure?
11. Define Knot Count metric.
12. Compare different software metrics.
13. What are the benefits of using code metrics?

UNIT 6: Software Testing and Quality Assurance

1. Define software testing.
2. Explain the psychology of testing.
3. What are the different levels of testing?
4. What are the steps of the testing process?
5. Explain the purpose of a test plan.
6. What is test case design?
7. What is black box testing?
8. Explain Boundary Value Analysis with example.
9. Explain Pairwise Testing and State Based Testing.
10. What is White Box Testing?
11. Describe test case generation and tool support.
12. Define Quality Assurance and Quality Control.
13. Explain the cost of quality in software projects.
14. What are software reviews?
15. Describe Software Quality Assurance (SQA) approaches.
16. What is software reliability?
17. Explain ISO 9000 and ISO 9001 standards for software quality.

UNIT 7: CASE Tools and System Dependability and Security

1. What are CASE tools?
2. Explain the importance of a Central Repository in CASE tools.
3. Differentiate between Upper CASE, Lower CASE, and Integrated CASE tools.
4. What is SCRUM? Describe its main roles and events.
5. What is a dependable system? Explain its attributes.
6. What is Reliability Engineering? Explain with metrics.
7. Explain Safety Engineering with examples.
8. Define Security Engineering.
9. What are the CIA principles of security?
10. What is Resilience Engineering? How does it differ from reliability?
11. Explain techniques to improve system dependability.
12. Write short notes on system reliability and recovery techniques.

UNIT 8: Advanced Software Engineering

1. Define Software Reuse. What are its benefits?
2. Explain Component Based Software Engineering (CBSE).
3. What is Distributed Software Engineering?
4. Explain Service Oriented Software Engineering (SOSE).
5. What is Real Time Software Engineering?
6. Differentiate between Hard and Soft Real Time systems.
7. Explain Systems Engineering with example.
8. What is a System of Systems (SoS)?
9. Explain the challenges in Systems of Systems development.
10. Discuss the future trends in Advanced Software Engineering.

---

materio.

This document represents a printable version of the resource. Please note that the content may have been updated since this version was printed. For the most recent and accurate edition, we encourage you to scan the QR code provided on the front page to access the online version.

All notes and materials are curated by **InsightRoom**, a knowledge initiative under **materio.** Content is compiled from trusted and credible sources to ensure the highest standards of accuracy and quality. Where applicable, AI powered tools such as Perplexity may be used to assist in gathering verified insights from across the web.

To explore more curated insights and educational resources, please visit:
https://materioa.netlify.app/blog

The InsightRoom