

# COMPUTER SCIENCE

## P-302 (SOFTWARE ENGINEERING)

1. Software engineering is a systematic approach to the development, operation, and maintenance of software. [True/False] True
2. Software engineering involves applying engineering principles to \_\_\_\_\_. Software
3. One of the main reasons to study software engineering is to \_\_\_\_\_ the quality of software products. Improve
4. Software engineering helps in managing the \_\_\_\_\_ associated with software projects. Complexity
5. In software engineering, the \_\_\_\_\_ model is often used as the basis for understanding software development processes. Waterfall
6. In the Classical Waterfall Model, the development process is divided into distinct \_\_\_\_\_. Phases
7. The first phase of the Classical Waterfall Model is \_\_\_\_\_, where the project's requirements are gathered and analyzed. Requirements
8. In the Classical Waterfall Model, once the requirements are gathered, the next phase is \_\_\_\_\_, where the system architecture is designed. Design
9. In the Classical Waterfall Model, the phase where actual coding and implementation take place is called \_\_\_\_\_. Implementation
10. In the Classical Waterfall Model, after coding, the software goes through \_\_\_\_\_ to detect and fix defects. Testing
11. In the Classical Waterfall Model, the phase where the software is delivered to the customer is called \_\_\_\_\_. Deployment
12. The Classical Waterfall Model is a \_\_\_\_\_ software development process. Sequential
13. The Iterative Waterfall Model breaks down the software development process into \_\_\_\_\_ iterations. Multiple
14. In the Iterative Waterfall Model, each iteration includes phases like requirements, design, implementation, testing, and \_\_\_\_\_. Refinement
15. The Iterative Waterfall Model allows for \_\_\_\_\_ and modification of software based on feedback from each iteration. Feedback
16. The Prototyping Model involves creating a \_\_\_\_\_ version of the software to understand and refine requirements. Prototype
17. In the Prototyping Model, the initial prototype helps in \_\_\_\_\_ user needs and requirements. Elicit
18. Evolutionary Models involve \_\_\_\_\_ and incremental development of the software. Incremental
19. The Evolutionary Model includes multiple \_\_\_\_\_, with each iteration building upon the previous one. Iterations
20. The Spiral Model is a risk-driven model that emphasizes \_\_\_\_\_ throughout the software development process. Risk management
21. In the Spiral Model, each iteration is referred to as a \_\_\_\_\_. Spiral
22. The Spiral Model consists of four major phases: \_\_\_\_\_, risk analysis, engineering, and evaluation. Planning
23. The \_\_\_\_\_ phase in the Spiral Model involves setting objectives, identifying alternatives, and evaluating risks. Planning

24. In the Spiral Model, the \_\_\_\_\_ phase focuses on actual software development and testing. Engineering
25. The Evolutionary Model is also known as the \_\_\_\_\_ model. Incremental
26. The Prototype Model helps in reducing \_\_\_\_\_ by clarifying user requirements early in the process. Uncertainty
27. The Classical Waterfall Model assumes that all requirements are known and stable from the \_\_\_\_\_. Beginning
28. The Iterative Waterfall Model allows for changes to be made in \_\_\_\_\_ iterations. Subsequent
29. In the Spiral Model, \_\_\_\_\_ is an essential aspect, ensuring that risks are managed throughout the project. Risk management
30. The main advantage of the Evolutionary Model is its ability to adapt to changing \_\_\_\_\_ during development. Requirements
1. Function Point Metric assesses a software project's \_\_\_\_\_ from a user's perspective. [functionality]
2. Project estimation techniques help in predicting the \_\_\_\_\_, effort, and resources required for a software project. [cost]
3. Empirical estimation techniques are based on \_\_\_\_\_ data from past projects. [historical]
4. Heuristic techniques rely on \_\_\_\_\_ and rules of thumb to make estimates. [experience]
5. Analytical estimation techniques use \_\_\_\_\_ models to estimate project parameters. [mathematical]
6. Expert judgment technique involves seeking input from \_\_\_\_\_ individuals with experience in the domain. [knowledgeable]
7. Delphi cost estimation is a technique that uses \_\_\_\_\_ to arrive at a consensus estimate. [experts]
8. COCOMO, which stands for CONstructive COSt MOdel, is a \_\_\_\_\_ estimation technique. [heuristic]
9. The Basic COCOMO model is a \_\_\_\_\_ estimation model. [simplified]
10. Intermediate COCOMO incorporates \_\_\_\_\_ factors into the estimation process. [scale]
11. Complete COCOMO is an elaboration of the \_\_\_\_\_ model. [basic]
12. COCOMO2 is an \_\_\_\_\_ version of the COCOMO model. [improved]
13. Function Point Metric considers factors such as \_\_\_\_\_ and data complexity to estimate project size. [processing]
14. In Lines of Code (LOC) estimation, a higher LOC count generally indicates a \_\_\_\_\_ project size. [larger]
15. Function points are calculated based on the \_\_\_\_\_ of inputs, outputs, and inquiries. [number]
16. Empirical estimation techniques use \_\_\_\_\_ data to make predictions. [historical]
17. Heuristic techniques often rely on \_\_\_\_\_ to make educated guesses. [rules]
18. Analytical estimation techniques utilize \_\_\_\_\_ equations to estimate project parameters. [mathematical]
19. Expert judgment technique relies on the experience of \_\_\_\_\_ individuals. [knowledgeable]
20. Delphi cost estimation seeks to achieve a consensus among a group of \_\_\_\_\_. [experts]
21. COCOMO estimates project \_\_\_\_\_ based on various factors. [effort]
22. The Basic COCOMO model provides a \_\_\_\_\_ estimate of project effort. [rough]
23. Intermediate COCOMO introduces \_\_\_\_\_ factors into the estimation process. [scale]
24. Complete COCOMO provides more \_\_\_\_\_ estimates than the basic model. [detailed]
25. COCOMO2 enhances the accuracy of \_\_\_\_\_ estimation. [cost]

26. In Function Point Metric, data complexity is assessed based on the number of \_\_\_\_\_. [file]
27. Empirical estimation techniques rely on \_\_\_\_\_ data from past projects. [historical]
28. Heuristic techniques use \_\_\_\_\_ and intuition to arrive at estimates. [experience]
29. Analytical estimation techniques employ \_\_\_\_\_ models to make predictions. [mathematical]

1. Lines of Code (LOC) is a commonly used metric for measuring the \_\_\_\_\_ of a software project. [size]
2. Function Point Metric assesses a software project's \_\_\_\_\_ from a user's perspective. [functionality]
3. Project estimation techniques help in predicting the \_\_\_\_\_, effort, and resources required for a software project. [cost]
4. Empirical estimation techniques are based on \_\_\_\_\_ data from past projects. [historical]
5. Heuristic techniques rely on \_\_\_\_\_ and rules of thumb to make estimates. [experience]
6. Analytical estimation techniques use \_\_\_\_\_ models to estimate project parameters. [mathematical]
7. Expert judgment technique involves seeking input from \_\_\_\_\_ individuals with experience in the domain. [knowledgeable]
8. Delphi cost estimation is a technique that uses \_\_\_\_\_ to arrive at a consensus estimate. [experts]
9. COCOMO, which stands for COConstructive COSt MOdel, is a \_\_\_\_\_ estimation technique. [heuristic]
10. The Basic COCOMO model is a \_\_\_\_\_ estimation model. [simplified]
11. Intermediate COCOMO incorporates \_\_\_\_\_ factors into the estimation process. [scale]
12. Complete COCOMO is an elaboration of the \_\_\_\_\_ model. [basic]
13. COCOMO2 is an \_\_\_\_\_ version of the COCOMO model. [improved]
14. Function Point Metric considers factors such as \_\_\_\_\_ and data complexity to estimate project size. [processing]
15. In Lines of Code (LOC) estimation, a higher LOC count generally indicates a \_\_\_\_\_ project size. [larger]
16. Function points are calculated based on the \_\_\_\_\_ of inputs, outputs, and inquiries. [number]
17. Empirical estimation techniques use \_\_\_\_\_ data to make predictions. [historical]
18. Heuristic techniques often rely on \_\_\_\_\_ to make educated guesses. [rules]
19. Analytical estimation techniques utilize \_\_\_\_\_ equations to estimate project parameters. [mathematical]
20. Expert judgment technique relies on the experience of \_\_\_\_\_ individuals. [knowledgeable]
21. Delphi cost estimation seeks to achieve a consensus among a group of \_\_\_\_\_. [experts]
22. COCOMO estimates project \_\_\_\_\_ based on various factors. [effort]
23. The Basic COCOMO model provides a \_\_\_\_\_ estimate of project effort. [rough]
24. Intermediate COCOMO introduces \_\_\_\_\_ factors into the estimation process. [scale]
25. Complete COCOMO provides more \_\_\_\_\_ estimates than the basic model. [detailed]
26. COCOMO2 enhances the accuracy of \_\_\_\_\_ estimation. [cost]
27. In Function Point Metric, data complexity is assessed based on the number of \_\_\_\_\_. [file]
28. Empirical estimation techniques rely on \_\_\_\_\_ data from past projects. [historical]
29. Heuristic techniques use \_\_\_\_\_ and intuition to arrive at estimates. [experience]
30. Analytical estimation techniques employ \_\_\_\_\_ models to make predictions. [mathematical]

1. Staffing level estimation involves determining the number of \_\_\_\_\_ required for a project. [staff]
2. Norden's work estimation model is based on the concept of \_\_\_\_\_. [man-hours]
3. Putnam's work estimation model considers the relationship between project \_\_\_\_\_ and project size. [effort]
4. Work Breakdown Structure (WBS) is a hierarchical representation of a project's \_\_\_\_\_. [tasks]
5. Activity networks and Critical Path Method (CPM) help in \_\_\_\_\_ the sequence and dependencies of project activities. [identifying]
6. Gantt charts provide a visual representation of project \_\_\_\_\_ and timelines. [schedules]
7. PERT charts use \_\_\_\_\_ to represent project activities and their dependencies. [nodes]
8. Risk management is the process of \_\_\_\_\_ and mitigating potential issues that could affect a project. [identifying]
9. Risk identification involves identifying \_\_\_\_\_ that could impact a project's success. [risks]

10. Risk assessment assesses the \_\_\_\_\_ and impact of identified risks. [probability]
11. Risk containment focuses on developing \_\_\_\_\_ to mitigate or respond to risks. [strategies]
12. Staffing level estimation helps in determining the appropriate number of \_\_\_\_\_ needed for a project. [personnel]
13. Norden's work estimation model is based on \_\_\_\_\_ required to complete a task. [man-hours]
14. Putnam's work estimation model considers the relationship between project \_\_\_\_\_ and effort. [size]
15. Work Breakdown Structure (WBS) breaks a project into smaller, manageable \_\_\_\_\_. [tasks]
16. Activity networks and Critical Path Method (CPM) help in identifying the \_\_\_\_\_ path of a project. [critical]
17. Gantt charts display project activities over a \_\_\_\_\_ timeline. [linear]
18. PERT charts use \_\_\_\_\_ to represent activities and dependencies. [arcs]
19. Risk management involves identifying, \_\_\_\_\_, and controlling risks. [assessing]
20. Risk identification is the process of \_\_\_\_\_ potential threats or opportunities. [recognizing]
21. Risk assessment involves evaluating the likelihood and \_\_\_\_\_ of identified risks. [impact]
22. Risk containment strategies are developed to \_\_\_\_\_ the effects of risks on a project. [mitigate]
23. Staffing level estimation helps in determining the appropriate \_\_\_\_\_ for a project. [resources]
24. Norden's work estimation model is based on the concept of \_\_\_\_\_ required for a task. [man-hours]
25. Putnam's work estimation model considers the relationship between project \_\_\_\_\_ and the effort needed. [size]
26. Work Breakdown Structure (WBS) provides a \_\_\_\_\_ view of project tasks. [hierarchical]
27. Activity networks and Critical Path Method (CPM) help in identifying the \_\_\_\_\_ path of a project. [critical]
28. Gantt charts visualize project activities over a \_\_\_\_\_ timeline. [linear]
29. PERT charts use \_\_\_\_\_ to represent activities and their dependencies. [nodes]
30. Risk management involves the process of \_\_\_\_\_ potential risks that may affect a project. [identifying]

#### SHORT TYPE

#### **Introduction to Software Engineering:**

##### **1. What is software engineering?**

- Software engineering is a systematic and disciplined approach to designing, developing, and maintaining software systems.

##### **2. Why study software engineering?**

- Studying software engineering helps produce high-quality software efficiently and economically.

#### **Software Life Cycle Models:**

##### **3. What is a software life cycle model?**

- A software life cycle model is a framework used to plan, design, build, test, and deliver software.

##### **4. What is the Classical Waterfall Model?**

- The Classical Waterfall Model is a linear and sequential software development model with distinct phases.

##### **5. Phases of the Classical Waterfall Model?**

- Phases include requirements, design, implementation, testing, deployment, and maintenance.

##### **6. What is the Iterative Waterfall Model?**

- It's a variant of the Waterfall Model where phases can be repeated or revisited.

##### **7. What is the Prototyping Model?**

- Prototyping involves building a working model to better understand and refine requirements.

##### **8. What is the Evolutionary Model?**

- The Evolutionary Model is an iterative approach that gradually builds the final product.

9. **What is the Spiral Model?**

- The Spiral Model combines iterative development with risk assessment.

**Metric for Project Size Estimation:**

10. **What is Lines of Code (LOC)?**

- LOC is a simple metric measuring software size based on the number of lines in the code.

11. **What is the Function Point Metric?**

- Function Points measure software size based on its functionality.

**Project Estimation Techniques:**

12. **What are Empirical Estimation Techniques?**

- Empirical techniques use historical data and experience for estimation.

13. **What are Heuristic Techniques?**

- Heuristic techniques rely on expert judgment and rules of thumb for estimation.

14. **What are Analytical Estimation Techniques?**

- Analytical techniques use mathematical models for estimation.

15. **What is Expert Judgment Technique?**

- Expert judgment involves seeking input from experienced individuals for estimation.

16. **What is Delphi Cost Estimation?**

- Delphi is a structured expert judgment technique using multiple rounds of feedback.

17. **What is COCOMO (Constructive Cost Model)?**

- COCOMO is a heuristic estimation technique for software development effort.

18. **What is the Basic COCOMO Model?**

- Basic COCOMO estimates effort based on lines of code and other factors.

19. **What is Intermediate COCOMO?**

- Intermediate COCOMO considers more project-specific factors for estimation.

20. **What is Complete COCOMO?**

- Complete COCOMO provides a detailed estimation process with various cost drivers.

21. **What is COCOMO2?**

- COCOMO2 is an updated version of COCOMO with improved accuracy.

**Staffing Level Estimation:**

22. **What is Norden's Work?**

- Norden's work introduced the concept of "staffing profiles" for estimating team size.

23. **What is Putnam's Work?**

- Putnam's work contributed to the development of effort estimation models.

**Work Breakdown Structure (WBS), Activity Networks, and Critical Path Method (CPM):**

24. **What is a Work Breakdown Structure (WBS)?**

- A WBS is a hierarchical decomposition of project tasks.

25. **What are Activity Networks?**

- Activity networks depict task dependencies in a project.

26. **What is the Critical Path Method (CPM)?**

- CPM identifies the longest path through a project's activities, determining project duration.

### **Gantt Charts and PERT Charts:**

#### **27. What is a Gantt Chart?**

- A Gantt Chart is a visual representation of project tasks over time.

#### **28. What is a PERT Chart?**

- A PERT Chart (Program Evaluation and Review Technique) is a tool for scheduling and managing projects.

### **Risk Management:**

#### **29. What is Risk Management in software engineering?**

- Risk management involves identifying, assessing, and mitigating risks in a project.

#### **30. What is Risk Identification?**

- Risk identification is the process of identifying potential risks in a project.

#### **31. What is Risk Assessment?**

- Risk assessment involves analyzing the likelihood and impact of identified risks.

#### **32. What is Risk Containment?**

- Risk containment strategies aim to minimize the impact of identified risks.

### **Software Design:**

#### **33. What is Cohesion and Coupling in software design?**

- Cohesion measures how closely related functions within a module are, while coupling measures interdependencies between modules.

#### **34. What is Function-Oriented Design?**

- Function-Oriented Design focuses on defining functions and their relationships.

#### **35. What are Data Flow Diagrams (DFDs)?**

- DFDs represent the flow of data within a system.

#### **36. What is Object-Oriented Design?**

- Object-Oriented Design structures software around objects and their interactions.

#### **37. What is Use Case Model Development?**

- Use case modeling defines system interactions from a user perspective.

### **Coding & Testing:**

#### **38. What is Black Box Test Suite Design Approach?**

- Black box testing focuses on testing software functionality without knowledge of internal code.

#### **39. What is White Box Testing?**

- White box testing examines the internal logic and code of a software application.

#### **40. What are Basic Concepts of Statement Coverage, Branch Coverage, Condition Coverage, and Path Coverage?**

- These are code coverage metrics used to assess the thoroughness of testing.

#### **41. What is McCabe's Cyclomatic Complexity Metric?**

- Cyclomatic complexity quantifies the complexity of a program's control flow.

#### **42. What is Data Flow-based Testing?**

- Data flow-based testing assesses how data is processed within a program.

#### **43. What is Mutation Testing?**

- Mutation testing introduces artificial defects into code to evaluate test coverage.

### **Introduction to Software Engineering:**



1. **Q:** What is software engineering? **A:** Software engineering is a systematic approach to designing, developing, and maintaining software systems.
2. **Q:** Why is software engineering important? **A:** Software engineering is important to ensure the quality, reliability, and maintainability of software systems.

### **Software Life Cycle Models:**

3. **Q:** What is a software life cycle model? **A:** A software life cycle model is a framework that defines the stages and activities involved in software development.
4. **Q:** What is the Classical Waterfall Model? **A:** The Classical Waterfall Model is a sequential software development model with distinct phases: requirements, design, implementation, testing, deployment, and maintenance.
5. **Q:** What are the phases of the Classical Waterfall Model? **A:** The phases are: requirements analysis, system design, coding, testing, integration, and maintenance.
6. **Q:** What is the Iterative Waterfall Model? **A:** The Iterative Waterfall Model is an extension of the Classical Waterfall Model where phases are revisited iteratively.
7. **Q:** What is the Prototyping Model? **A:** The Prototyping Model involves creating a preliminary version of the software to gather user feedback and refine requirements.
8. **Q:** What is the Evolutionary Model? **A:** The Evolutionary Model develops the software incrementally, with each increment adding new features or improvements.
9. **Q:** What is the Spiral Model? **A:** The Spiral Model combines iterative development with risk assessment at each iteration.

### **Metric for Project Size Estimation:**

10. **Q:** What is Lines of Code (LOC)? **A:** LOC is a metric that measures the size of a software project by counting the number of lines of code in the source code.
11. **Q:** What is the Function Point Metric? **A:** Function Point Metric is a size estimation technique that quantifies the functionality of a software application.

### **Project Estimation Techniques:**

12. **Q:** What are Empirical Estimation Techniques? **A:** Empirical estimation techniques use historical data and experience to estimate project effort and duration.
13. **Q:** What are Heuristic Techniques? **A:** Heuristic techniques use rules of thumb and best practices for project estimation.
14. **Q:** What are Analytical Estimation Techniques? **A:** Analytical estimation techniques use mathematical models to estimate project parameters.
15. **Q:** What is Expert Judgment Technique? **A:** Expert judgment technique involves seeking input from experienced individuals for project estimation.
16. **Q:** What is Delphi Cost Estimation? **A:** Delphi Cost Estimation is a consensus-based estimation technique where experts provide anonymous estimates that are then discussed and refined.
17. **Q:** What is COCOMO (Constructive Cost Model)? **A:** COCOMO is a heuristic estimation technique that estimates effort, cost, and duration based on project characteristics.
18. **Q:** What is the Basic COCOMO Model? **A:** Basic COCOMO estimates effort based on project size and complexity.
19. **Q:** What is Intermediate COCOMO? **A:** Intermediate COCOMO adds additional cost drivers to the Basic COCOMO model for more accurate estimates.
20. **Q:** What is Complete COCOMO? **A:** Complete COCOMO is an advanced version of COCOMO that considers a wide range of factors for estimation.
21. **Q:** What is COCOMO2? **A:** COCOMO2 is an enhanced version of COCOMO that incorporates modern software development practices.

### **Staffing Level Estimation:**

22. **Q:** What is Norden's Work? **A:** Norden's Work is a model for estimating the number of software engineers required for a project based on project size.

23. **Q:** What is Putnam's Work? **A:** Putnam's Work is a model for estimating project effort and duration based on historical data.

### **Work Breakdown Structure and Project Scheduling:**

24. **Q:** What is a Work Breakdown Structure (WBS)? **A:** A WBS is a hierarchical decomposition of a project into tasks, subtasks, and work packages.

25. **Q:** What are Activity Networks and Critical Path Method (CPM)? **A:** Activity networks and CPM are techniques for scheduling and managing project activities.

26. **Q:** What are Gantt Charts? **A:** Gantt Charts are visual representations of project schedules, showing tasks and their durations.

27. **Q:** What are PERT Charts? **A:** PERT (Program Evaluation and Review Technique) Charts are used for project scheduling, accounting for task uncertainties.

### **Risk Management:**

28. **Q:** What is Risk Identification? **A:** Risk identification is the process of identifying potential threats and opportunities that may impact a project.

29. **Q:** What is Risk Assessment? **A:** Risk assessment involves analyzing the likelihood and impact of identified risks.

30. **Q:** What is Risk Containment? **A:** Risk containment involves developing strategies to mitigate or manage identified risks.

### **Software Design:**

31. **Q:** What is Cohesion and Coupling in software design? **A:** Cohesion refers to how closely related the elements of a module or component are, while coupling refers to the interdependence between modules or components.

32. **Q:** What is Function-Oriented Design? **A:** Function-Oriented Design focuses on decomposing a system into functions and data.

33. **Q:** What are Data Flow Diagrams (DFDs)? **A:** DFDs are graphical representations of how data flows within a system.

34. **Q:** What is Object-Oriented Design? **A:** Object-Oriented Design involves modeling a system using objects, classes, and their relationships.

35. **Q:** What is Use Case Model Development? **A:** Use Case Model Development identifies and describes the interactions between users and a system.

### **Coding and Testing:**

36. **Q:** What is Black Box Test Suite Design Approach? **A:** Black Box Testing focuses on testing the software's functionality without knowledge of its internal structure.

37. **Q:** What is White Box Testing? **A:** White Box Testing examines the internal logic and structure of the software.

38. **Q:** What is Statement Coverage in White Box Testing? **A:** Statement Coverage measures the percentage of code statements executed during testing.

39. **Q:** What is Branch Coverage in White Box Testing? **A:** Branch Coverage measures the percentage of decision branches covered during testing.

40. **Q:** What is Condition Coverage in White Box Testing? **A:** Condition Coverage ensures that all possible conditions in code are tested.

41. **Q:** What is Path Coverage in White Box Testing? **A:** Path Coverage aims to test all possible execution paths through the code.

42. **Q:** What is McCabe's Cyclomatic Complexity Metric? **A:** McCabe's Cyclomatic Complexity is a metric used to measure the complexity of a software program's control flow.



43. **Q:** What is Data Flow-based Testing? **A:** Data Flow-based Testing checks the flow of data through a program to identify issues.

44. **Q:** What is Mutation Testing? **A:** Mutation Testing involves introducing small changes (mutations) to the code to assess the test suite's effectiveness.

LONG TYPE

### **Introduction to Software Engineering:**

1. What is software engineering, and why is it important in the field of technology?
2. How has the field of software engineering evolved over the years?
3. What are the key principles of software engineering?
4. What role does software engineering play in modern businesses?
5. Explain the significance of software engineering in software development.
6. How does software engineering differ from traditional programming?
7. What are the core objectives of software engineering?
8. What are the key challenges faced by software engineers today?
9. How does software engineering contribute to the development of robust and reliable software?
10. What are some ethical considerations in software engineering?

### **Why Study Software Engineering:**

11. Why should individuals pursue a career in software engineering?
12. How does a formal education in software engineering benefit aspiring professionals?
13. What are the potential career paths for software engineers?
14. What industries rely heavily on software engineering expertise?
15. How can software engineering skills contribute to solving real-world problems?
16. What is the future outlook for jobs in software engineering?
17. What personal qualities are essential for success in software engineering?
18. How does software engineering impact innovation in technology?
19. What role does continuous learning play in the career of a software engineer?
20. How does software engineering foster creativity and problem-solving?

### **Software Life Cycle Models:**

21. What is a software development life cycle (SDLC)?
22. Describe the importance of selecting an appropriate SDLC model.
23. Compare and contrast the various SDLC models.
24. How does the choice of an SDLC model affect project outcomes?
25. Explain the stages involved in the software development life cycle.
26. What are the advantages of using a structured SDLC model?
27. What are the disadvantages of using a rigid SDLC model?
28. How do iterative SDLC models differ from sequential models?
29. Give examples of when to use agile SDLC models.
30. What considerations should be made when tailoring an SDLC model for a specific project?

### **Classical Waterfall Model:**

31. What is the classical waterfall model in software engineering?
32. Describe the phases of the classical waterfall model.
33. What are the key characteristics of each phase in the waterfall model?
34. What are the advantages of using the waterfall model?
35. What are the limitations of the waterfall model?
36. How does the waterfall model handle changes in project requirements?

37. Is the waterfall model suitable for all types of software projects?
38. Explain the importance of documentation in the waterfall model.
39. What are the key success factors for a project following the waterfall model?
40. How can risk management be integrated into the waterfall model?

#### **Phases of Classical Waterfall Model:**

41. What is the Requirements phase in the classical waterfall model?
42. How does the Requirements phase ensure a clear understanding of the project's goals?
43. Describe the Design phase in the waterfall model and its objectives.
44. How does the Design phase translate requirements into a concrete plan?
45. What happens during the Implementation phase in the classical waterfall model?
46. How is the Implementation phase different from coding?
47. Explain the Testing phase in the context of the waterfall model.
48. What are the key objectives of the Testing phase?
49. How does the Maintenance phase contribute to the software's long-term success?
50. What is the significance of the Documentation phase in the waterfall model?

#### **Iterative Waterfall Model:**

51. How does the iterative waterfall model address the limitations of the classical waterfall model?
52. What is the primary concept behind iterative development in software engineering?
53. Describe the iterative phases in the iterative waterfall model.
54. What role does feedback play in the iterative waterfall model?
55. How can iterative development improve project flexibility and adaptability?
56. What are the potential challenges of using the iterative waterfall model?
57. Give examples of projects that are well-suited for the iterative waterfall approach.
58. How is risk management integrated into the iterative waterfall model?
59. Explain how the iterative waterfall model handles changes in requirements.
60. What is the role of continuous improvement in iterative development?

#### **Prototyping Model:**

61. What is the prototyping model in software engineering?
62. How does the prototyping model differ from the traditional waterfall model?
63. Describe the key phases of the prototyping model.
64. What is the purpose of creating a prototype in software development?
65. What types of projects benefit most from the prototyping approach?
66. What are the limitations of the prototyping model?
67. How can user feedback influence the prototyping process?
68. Explain the role of rapid prototyping in software development.
69. What happens after the prototype is approved in the prototyping model?
70. How can the prototyping model save time and resources in a project?

#### **Evolutionary Model:**

71. What is the evolutionary model in software engineering?
72. Describe the key characteristics of the evolutionary model.
73. How does the evolutionary model handle changing requirements?
74. What are the advantages of incremental development in the evolutionary model?
75. What is the role of user feedback in the evolutionary model?
76. How can the evolutionary model accommodate the growth of a software project?
77. What types of projects are best suited for the evolutionary approach?
78. Explain how the evolutionary model ensures early and continuous delivery.
79. What are the potential challenges of implementing the evolutionary model?

80. How does the evolutionary model impact software maintenance?

**Spiral Model:**

81. What is the spiral model in software engineering?

82. Describe the key phases of the spiral model.

83. How does risk management play a central role in the spiral model?

84. What are the advantages of using the spiral model?

85. What are the limitations of the spiral model?

86. How does the spiral model handle evolving requirements?

87. Explain the concept of iterations in the spiral model.

88. How is user involvement managed in the spiral model?

89. What types of projects benefit most from the spiral approach?

90. How does the spiral model balance risk and reward in a project?

**Metric for Project Size Estimation:**

91. What is the purpose of project size estimation in software engineering?

92. Describe the Lines of Code (LOC) metric for project size estimation.

93. What are the advantages of using LOC as a size metric?

94. What are the limitations of relying solely on LOC for project estimation?

95. Explain the Function Point Metric for project size estimation.

96. How does the Function Point Metric consider software complexity?

97. What are the advantages of using Function Points in project estimation?

98. How do project estimation techniques help in resource allocation?

99. What is the significance of size estimation in project planning?

100. How can software size metrics be used to improve project management?