

Section -A

- 1 . Can you name the common software development life cycle models? (CO:I)(BL:I)
- 2 . Where can you find reliable resources for keeping up with the latest software development trends?(CO:I)(BL:I)
- 3 . What is the fundamental nature of software and explain how they guide the development of complex software systems. (CO:I)(BL:I)
- 4 . How has the changing the nature of software evolved over the years, and what are the key trends shaping its future? (CO:I)(BL:I)
- 5 . Classify the different stages of the software development process. (CO:I)(BL:II)
- 6 . Classify different types of software process models. (CO:I)(BL:II)
- 7 . Compare the Waterfall and Spiral process models in software development. (CO:I)(BL:II)
- 8 . Outline the major steps in the Agile process model. (CO:I)(BL:II)
- 9 . Apply the principles of the Evolutionary Process Model to improve project adaptability. (CO:I)(BL:III)
- 10 . Experiment with different tools and technologies to facilitate concurrent development. (CO:I)(BL:III)
- 11 . Apply the principles of Agile development to a specific project to enhance its adaptability. (CO:I)(BL:III)
- 12 . Analyze the key principles and values underlying Extreme Programming. (CO:I)(BL:IV)
- 13 . List the primary principles and practices of Feature Driven Development. (CO:I)(BL:IV)
- 14 . Discover real-world examples of successful projects that embraced the principles of the Transition phase in UP? (CO:I)(BL:IV)
- 15 . How would you define the specific software requirement in the context of the project? (CO:II)(BL:I)
- 16 . How does the software requirement contribute to the overall functionality and objectives of the project? (CO:II)(BL:I)
- 17 . What are the essential components and functionalities of the software requirement? (CO:II)(BL:I)
- 18 . Recall any challenges or successes from previous projects related to requirements engineering? (CO:II)(BL:I)
- 19 . Classify the different types or categories of requirements that are relevant to the project? (CO:II)(BL:II)
- 20 . Outline the step-by-step process of translating customer requirements into engineering characteristics using QFD? (CO:II)(BL:II)
- 21 . Interpret the relationships between different use cases and their dependencies within the system? (CO:II)(BL:II)
- 22 . Demonstrate how a specific set of requirements aligns with the overall goals and objectives of the project? (CO:II)(BL:II)
- 23 . Develop a collaborative environment that encourages open communication and constructive dialogue during requirements negotiation? (CO:II)(BL:III)
- 24 . Interviews with stakeholders to validate their understanding and expectations related to the requirements? (CO:II)(BL:III)

- 25 . Select validation criteria and metrics to measure the success and effectiveness of the validation process? (CO:II)(BL:III)
- 26 . Analyze the gathered requirements to identify patterns, dependencies, and key features for building the requirements model? (CO:II)(BL:IV)
- 27 . Categorize the identified requirements into different groups or types to facilitate the building of a structured requirements model? (CO:II)(BL:IV)
- 28 . Conclusions can be drawn from the requirements elicitation process, and how do they guide the next steps in the project? (CO:II)(BL:IV)
- 29 . Choose the specific analysis methods and techniques to be applied in the project? (CO:III)(BL:I)
- 30 . Tell the story of how the design evolved and was refined throughout different iterations? (CO:III)(BL:I)
- 31 . Show examples or visual representations of the scenario-based models created during this phase? (CO:III)(BL:I)
- 32 . Why were specific scenarios chosen for modeling, and what criteria or considerations influenced these decisions? (CO:III)(BL:I)
- 33 . Demonstrate specific instances where the data model effectively represented and organized the underlying information? (CO:III)(BL:II)
- 34 . Demonstrate specific instances where data objects were effectively utilized in the system or project? (CO:III)(BL:II)
- 35 . Illustrate how different classes and their relationships were visually represented in the class-based model? (CO:III)(BL:II)
- 36 . Summary that captures the main insights gained from the identification and utilization of classes in the model? (CO:III)(BL:II)
- 37 . Construct the flow-oriented model to ensure clarity, accuracy, and alignment with project goals? (CO:III)(BL:III)
- 38 . Planning activities were undertaken to ensure the success of creating a scenario-based model that meets project objectives? (CO:III)(BL:III)
- 39 . Planning activities were undertaken to ensure that the overall objectives and philosophy were effectively communicated and understood by all stakeholders? (CO:III)(BL:III)
- 40 . Assumptions were made during the design process, and how did they influence design decisions? (CO:III)(BL:IV)
- 41 . list the key components or elements within the design model that play a crucial role in the overall system design? (CO:III)(BL:IV)
- 42 . Relationships between different elements within the design model considered and optimized? (CO:III)(BL:IV)
- 43 . Relate the concept of "boundary value analysis" to software testing and explain its importance. (CO:IV)(BL:I)
- 44 . Recall and explain the principles of the V-Model in the context of software testing strategy. (CO:IV)(BL:I)
- 45 . How can a test-driven development (TDD) approach contribute to an effective software testing strategy? (CO:IV)(BL:I)
- 46 . What are the key benefits of implementing continuous integration in the software testing process? (CO:IV)(BL:I)
- 47 . Summarize the goals and objectives of system testing in three key points. (CO:IV)(BL:II)
- 48 . Demonstrate the importance of system testing in ensuring the overall reliability and performance of a software application. (CO:IV)(BL:II)

- 49 . Compare system testing with acceptance testing, highlighting the key differences and similarities. (CO:IV)(BL:II)
- 50 . Contrast black-box testing and white-box testing in the context of system testing, discussing their distinct approaches. (CO:IV)(BL:II)
- 51 . Apply a systematic debugging approach to resolve a specific error in a given code snippet. (CO:IV)(BL:III)
- 52 . Build a comprehensive guide on choosing and implementing effective debugging strategies for various types of software bugs. (CO:IV)(BL:III)
- 53 . Interview a software developer to understand their perspective on the collaboration between developers and testers in white-box testing processes (CO:IV)(BL:III)
- 54 . List the key components that should be included in a black-box testing plan for a large-scale e-commerce application. (CO:IV)(BL:IV)
- 55 . Discover common security concerns in web-based applications and propose strategies to address these concerns using object-oriented principles. (CO:IV)(BL:IV)
- 56 . Analyze the impact of responsive design on user interface testing and discuss strategies for ensuring consistency across different devices. (CO:IV)(BL:IV)
- 57 . List the common challenges faced in software project management and propose strategies to overcome these challenges. (CO:V)(BL:I)
- 58 . List the essential project metrics that should be included in a project dashboard to provide an overall view of project health. (CO:V)(BL:I)
- 59 . Label the different categories of metrics used in software project management (e.g., productivity metrics, quality metrics) and discuss their unique contributions. (CO:V)(BL:I)
- 60 . Name the key components of a Project Charter in software project management, detailing their significance in project initiation. (CO:V)(BL:I)
- 61 . Translate high-level project requirements into meaningful estimates, breaking down the estimation process into manageable tasks and milestones. (CO:V)(BL:II)
- 62 . Classify different types of risks that can impact project scheduling, distinguishing between internal and external factor. (CO:V)(BL:II)
- 63 . Explain the concept of risk response strategies in project scheduling and how they can be applied to mitigate the impact of identified risks. (CO:V)(BL:II)
- 64 . Illustrate the use of a risk register in project scheduling, showcasing how it captures and tracks identified risks, their impact, and proposed responses. (CO:V)(BL:II)
- 65 . Interview team members to gather insights on their experiences with SCM processes, identifying areas for improvement and optimization. (CO:V)(BL:III)
- 66 . Utilize project metrics to conduct a post-project analysis, extracting lessons learned and identifying areas for improvement in future projects. (CO:V)(BL:III)
- 67 . Plan the implementation of automated metrics collection tools, streamlining the process of gathering and analyzing project data. (CO:V)(BL:III)
- 68 . Discover key factors that contribute to team motivation and productivity in software project management. (CO:V)(BL:IV)
- 69 . Analyze the impact of improper version control on software development projects and propose strategies for improvement. (CO:V)(BL:IV)
- 70 . List common challenges in software project estimation and discuss mitigation strategies for each challenge. (CO:V)(BL:IV)

Section -B

1. How would you define the role of version control in software development? (CO:I)(BL:I)
2. Match different software development methodologies with their respective characteristics. (CO:I)(BL:I)

3. How can project managers select the most suitable software development methodology for a given project? (CO:I)(BL:I)
4. What are the key characteristics of a well-designed software interface? (CO:I)(BL:I)
5. Translate the principles of the Waterfall model into practical steps. (CO:I)(BL:II)
6. Rephrase the principles of the Rational Unified Process (RUP) for better understanding. (CO:I)(BL:II)
7. Illustrate the key phases of the Unified Process (UP) model. (CO:I)(BL:II)
8. illustrate how the Big Bang model works in a real-world software development scenario? (CO:I)(BL:II)
9. Build a case study demonstrating the successful implementation of the Evolutionary Process Model. (CO:I)(BL:III)
10. Apply the principles of the Evolutionary Process Model to improve project adaptability. (CO:I)(BL:III)
11. Build a case study illustrating the successful implementation of concurrent models in a large-scale project. (CO:I)(BL:III)
12. Categorize the core practices of Extreme Programming into planning, coding, and testing phases. (CO:I)(BL:IV)
13. Dissect the role of continuous learning in Adaptive Software Development and its impact on software development teams. (CO:I)(BL:IV)
14. Test for the impact of regular feature list development on project predictability in Feature Driven Development. (CO:I)(BL:IV)
15. What methods or tools will be used to find and gather information about the software requirements? (CO:II)(BL:I)
16. list the key features and functionalities associated with the software requirement? (CO:II)(BL:I)
17. What are the essential principles and best practices of requirements engineering that should be followed? (CO:II)(BL:I)
18. How does the requirements engineering process contribute to the overall success and quality of the project? (CO:II)(BL:I)
19. Explain the rationale behind the selection and prioritization of specific engineering characteristics in relation to customer needs?(CO:II)(BL:II)
20. Illustrate the relationships between customer requirements and engineering characteristics using QFD matrices or diagrams? (CO:II)(BL:II)
21. Translate the requirements model into actionable tasks for the development, testing, and implementation phases? (CO:II)(BL:II)
22. Show examples where the application of analysis patterns has led to successful software design and development outcomes? (CO:II)(BL:II)
23. Identify key stakeholders and their perspectives to ensure a comprehensive understanding of their requirements? (CO:II)(BL:III)
24. Utilize feedback from stakeholders and the results of validation activities to refine and improve the requirements? (CO:II)(BL:III)
25. Make use of validation tools, such as automated testing or analysis tools, to enhance the accuracy and efficiency of the process (CO:II)(BL:III)
26. Survey among stakeholders to validate and refine software requirements based on their feedback? (CO:II)(BL:IV)
27. Dissect complex stakeholder needs into specific, detailed requirements for more effective elicitation? (CO:II)(BL:IV)
28. Inferences can be made about unspoken requirements based on stakeholder behaviors, preferences, and industry trends? (CO:II)(BL:IV)

29. How did the team approach the analysis of requirements and system components? (CO:III)(BL:I)
30. Where were design decisions documented, stored, and made accessible to the project team? (CO:III)(BL:I)
31. Which stakeholders were involved in the scenario-based modeling decisions, and how was their input considered? (CO:III)(BL:I)
32. How did the team approach the creation and representation of scenarios in the modeling process (CO:III)(BL:I)
33. Show examples or visual representations of the data model, including entity-relationship diagrams or other relevant artifacts? (CO:III)(BL:II)
34. Rephrase complex data object concepts to enhance understanding among non-technical stakeholders? (CO:III)(BL:II)
35. Summary that captures the main insights gained from the identification and utilization of data attributes? (CO:III)(BL:II)
36. Show examples or visual representations of how data attributes were utilized in the system, such as entity-attribute diagrams or other relevant artifacts? (CO:III)(BL:II)
37. Development of the flow-oriented model integrated into the overall project development process? (CO:III)(BL:III)
38. Experimental approaches or pilot studies conducted during the creation of the flow-oriented model, and what insights were gained? (CO:III)(BL:III)
39. Construct the scenario-based model to ensure it effectively represented real-world scenarios and user interactions? (CO:III)(BL:III)
40. Conclusions were drawn from the design analysis and how were they used to shape the overall design approach? (CO:III)(BL:IV)
41. Inferences were drawn from the analysis of user needs, business requirements, and technical constraints during the design process? (CO:III)(BL:IV)
42. Testing methodologies were employed to test and validate the effectiveness of the design model? (CO:III)(BL:IV)
43. When in the software development life cycle is acceptance testing typically performed, and why? (CO:IV)(BL:I)
44. Why is it important to conduct performance testing for software applications, and what key metrics are usually measured in this type of testing? (CO:IV)(BL:I)
45. Select and compare two testing strategies (e.g., top-down testing vs. bottom-up testing) based on their advantages and limitations. (CO:IV)(BL:I)
46. Tell how a risk-based testing strategy can be aligned with business priorities and project goals. (CO:IV)(BL:I)
47. Classify the different types of system testing and explain the specific objectives of each. (CO:IV)(BL:II)
48. Compare system testing with acceptance testing, highlighting the key differences and similarities. (CO:IV)(BL:II)
49. Explain the role of system testing in the software development life cycle and its significance in detecting defects. (CO:IV)(BL:II)
50. Show the step-by-step process of executing a test case during the system testing phase. (CO:IV)(BL:II)
51. Choose a debugging tool that you find most effective for a specific type of issue and explain your selection. (CO:IV)(BL:III)
52. Utilize automated testing frameworks as part of the debugging process to catch and prevent bugs earlier in the development cycle. (CO:IV)(BL:III)
53. Experiment with mutation testing techniques in a white-box testing context, identifying how it helps assess the effectiveness of test cases. (CO:IV)(BL:III)

54. Analyze the advantages and disadvantages of using equivalence partitioning in black-box testing and provide examples. (CO:IV)(BL:IV)
55. Compare black-box testing and white-box testing methodologies, highlighting their differences in terms of approach and objectives. (CO:IV)(BL:IV)
56. Examine the role of user personas in black-box testing and how they contribute to the creation of realistic test scenarios. (CO:IV)(BL:IV)
57. Choose a software project management methodology (e.g., Agile, Waterfall, Scrum) and discuss its suitability for different types of projects. (CO:V)(BL:I)
58. Find examples of successful software projects and analyze the project management practices that contributed to their success. (CO:V)(BL:I)
59. How does risk management play a crucial role in software project management, and what strategies can be employed to mitigate potential risks? (CO:V)(BL:I)
60. Choose key performance indicators (KPIs) for measuring software development productivity and explain their relevance. (CO:V)(BL:I)
61. Compare the advantages and disadvantages of top-down and bottom-up estimation approaches in software project management. (CO:V)(BL:II)
62. Summarize the key considerations when estimating software project duration and cost, including the importance of involving key stakeholders. (CO:V)(BL:II)
63. Compare the effectiveness of qualitative risk analysis versus quantitative risk analysis in the context of project scheduling. (CO:V)(BL:II)
64. Contrast the approaches to project scheduling when dealing with known risks versus unknown risks, highlighting the importance of contingency planning. (CO:V)(BL:II)
65. Develop a strategy for managing configuration changes in a dynamic software environment, emphasizing the importance of traceability and impact analysis. (CO:V)(BL:III)
66. Identify key elements in a software configuration baseline, discussing how they provide a snapshot of a project's state at a specific point in time.(CO:V)(BL:III)
67. Model a continuous improvement culture within a development team, emphasizing the iterative nature of SPI and the importance of ongoing feedback.(CO:V)(BL:III)
68. List key stakeholders in a software project and discuss their influence on project outcomes. (CO:V)(BL:IV)
69. Explore the motive behind conducting a feasibility study before the initiation of a software project. (CO:V)(BL:IV)
70. Conclusion on the importance of revisiting and refining project estimates throughout the software development life cycle. (CO:V)(BL:IV)

Section -C

- 1 . List and elaborate on the advantages and disadvantages encountered while applying the Spiral Model in your case study. (CO:I)(BL:I)
- 2 . Illustrate the key phases and activities of the software processes in your case study with graphical representations or flowcharts. (CO:I)(BL:II)
- 3 . Identify and explain the major milestones in the Waterfall Model. (CO:I)(BL:III)
- 4 . Interview a project team following the Waterfall Model, what key aspects would you inquire about in each phase? (CO:I)(BL:III)
- 5 . Apply the Incremental Process Model in a software development project? (CO:I)(BL:III)
- 6 . Categorize the features in a software project based on their priority and dependencies within the context of FDD? (CO:I)(BL:IV)
- 7 . Compare Feature Driven Development with other software development methodologies. (CO:I)(BL:IV)
- 8 . Tell the story of how software requirements evolved and were refined throughout the different phases of the case study? (CO:II)(BL:I)

- 9 . Rephrase or simplify technical aspects of the requirements engineering process as described in the case study to enhance clarity? (CO:II)(BL:II)
- 10 . Planning activities were undertaken to ensure the success of the requirements elicitation process in terms of coverage and stakeholder satisfaction? (CO:II)(BL:III)
- 11 . Develop a comprehensive plan for requirements elicitation, considering various aspects such as timelines, resources, and stakeholder availability? (CO:II)(BL:III)
- 12 . List the key elements or components that were included in the Requirements Model based on the analyzed requirements in the case study? (CO:II)(BL:IV)
- 13 . Motives or goals underlay the inclusion of specific requirements in the Requirements Model, and how did these motives align with the project's objectives? (CO:II)(BL:IV)
- 14 . Inferences can be made about the impact of effective negotiation on the overall success of the project in the case study? (CO:II)(BL:IV)
- 15 . Can you list the main scenarios explored in the case study and their respective characteristics? (CO:III)(BL:I)
- 16 . Show examples of data models created during different phases of the case study, and explain their significance? (CO:III)(BL:II)
- 17 . Construction of flow-oriented models approached, and what methodologies or tools were utilized in the process? (CO:III)(BL:III)
- 18 . Interviews conducted with stakeholders or subject matter experts to gather insights for constructing accurate flow-oriented models? (CO:III)(BL:III)
- 19 . Development of data design elements contribute to achieving the overall goals and objectives of the system in the case study? (CO:III)(BL:III)
- 20 . Assumptions made during the architectural design phase, and how did they influence the selection of design elements? (CO:III)(BL:IV)
- 21 . Conclusions were drawn from the analysis of architectural design options, and how did they inform the final design decision in the case study? (CO:III)(BL:IV)
- 22 . Name the tools and technologies used for automated white-box testing in the case study, discussing their features and benefits. (CO:IV)(BL:I)
- 23 . Compare the effectiveness of automated black-box testing tools with manual testing efforts in the case study, highlighting advantages and limitations. (CO:IV)(BL:II)
- 24 . Plan the execution of a performance testing strategy for a mobile application in the case study, considering the unique challenges and constraints of mobile platforms.(CO:VI)(BL:II)
- 25 . Identify performance bottlenecks in the case study, using performance testing results to pinpoint areas of the application that may need optimization.(CO:IV)(BL:III)
- 26 . Examine the role of threat modeling in the security testing process, discussing how it helps identify and prioritize potential risks. (CO:IV)(BL:IV)
- 27 . Compare the advantages and disadvantages of automated testing versus manual testing strategies, considering factors like scalability and accuracy. (CO:IV)(BL:IV)
- 28 . Analyze the effectiveness of risk-based testing strategies in mitigating project risks, and discuss scenarios where it is most beneficial. (CO:IV)(BL:IV)
- 29 . Tell the story of a software project that faced significant challenges and discuss the project management strategies employed to bring it to a successful conclusion.(CO:V)(BL:I)
- 30 . Explain the appropriate project management metrics for tracking the progress of a software project, considering factors such as team productivity and project timelines. (CO:V)(BL:II)
- 31 . Make use of root cause analysis techniques in SPI to identify underlying issues contributing to recurrent problems in software development. (CO:V)(BL:III)
- 32 . Plan the implementation of a new software development methodology (e.g., DevOps) as part of SPI, detailing the steps and expected benefits. (CO:V)(BL:III)

33 . Take part in a risk identification workshop and discuss the importance of involving project stakeholders in the process. (CO:V)(BL:IV)

34 . Divide a software project into phases and discuss the importance of each phase in the Software Development Life Cycle (SDLC). (CO:V)(BL:IV)

35 . List common challenges in software project estimation and discuss mitigation strategies for each challenge. (CO:V)(BL:IV)