Here are brief answers to your interview questions:

1. **What do you understand by software testing?** Software testing is the process of evaluating and verifying that a software application or system functions as expected and meets the required standards.
2. **When should you stop the testing process?** Testing should stop when all critical test cases are executed, the desired quality levels are achieved, and there is sufficient confidence that the software is ready for production.
3. **What do verification and validation mean in software testing?** Verification ensures the software is built according to the requirements, while validation ensures the software meets user needs and performs as expected.
4. **What is static testing? When does it start and what does it cover?** Static testing involves reviewing and analyzing code, requirements, and design without executing the program. It starts early in the development cycle and covers code reviews, walkthroughs, and inspections.
5. **Define black-box testing.** Black-box testing evaluates the functionality of the software without knowing its internal workings. It focuses on input-output behavior.
6. **What is a test plan and what does it include?** A test plan outlines the testing approach, objectives, scope, resources, schedule, and deliverables. It includes test strategies, test environment, and risk analysis.
7. **What is Traceability Matrix?** A Traceability Matrix maps requirements to test cases, ensuring each requirement is covered by a test.
8. **What is meant by test coverage?** Test coverage measures the extent to which the code or functionality is tested. It ensures that all code paths or requirements are tested.
9. **Is it possible to achieve 100% testing coverage? How would you ensure it?** Achieving 100% testing coverage is difficult but can be ensured by combining various testing techniques like unit testing, integration testing, and code coverage analysis tools.
10. **What are unit testing and integration testing?** Unit testing focuses on testing individual components of the software, while integration testing evaluates how different components work together.
11. **Can we do system testing at any stage?** No, system testing is done after integration testing, when all components are integrated and the system is stable.
12. **Mention the different types of software testing.** Types include functional, non-functional, regression, performance, security, usability, unit, integration, system, acceptance, and more.
13. **What is the difference between a test driver and a test stub?** A test driver is used to simulate the behavior of a component that calls the unit being tested, while a test stub simulates the behavior of a called unit.
14. **What is agile testing and why is it important?** Agile testing is a flexible and iterative approach to testing in agile development. It ensures continuous feedback, adaptation, and quicker release cycles.
15. **What do you know about data flow testing?** Data flow testing focuses on how data moves through the software, ensuring proper initialization, usage, and deletion of data variables.
16. **What is the purpose of the end-to-end testing?** End-to-end testing validates the complete workflow of the application, ensuring that all integrated components work together as expected.
17. **Can you explain the importance of test cases in the manual testing process?** Test cases provide clear instructions on how to verify the software’s functionality, ensuring that tests are repeatable and consistent.
18. **What is regression testing, and why is it important?** Regression testing ensures that new changes don’t negatively affect existing functionality. It's crucial for maintaining software stability after updates.
19. **How do you prioritize test cases when you have limited time for testing?** Prioritize critical functionalities, high-risk areas, and commonly used features, ensuring that the most impactful tests are covered first.
20. **The probability that a server-class application hosted on the cloud is up and running for six long months without crashing is 99.99 percent. To analyze this type of scenario, what test will you perform?** Perform **reliability testing** and simulate system failures to ensure uptime meets the expected standards.
21. **What will you do when a bug turns up during testing?** Report the bug with clear details, including steps to reproduce, and track it until it’s resolved.
22. **How do you test a product if the requirements are yet to be freezed?** Perform exploratory testing, create test cases based on assumptions, and validate any available partial requirements.
23. **If a product is in the production stage and one of its modules gets updated, then is it necessary to perform regression testing?** Yes, regression testing is necessary to ensure that the update doesn’t break any existing functionality.
24. **How will you overcome the challenges faced due to the unavailability of proper documentation for testing?** Collaborate with developers and stakeholders, create test cases based on available code, and perform exploratory testing.
25. **What do you understand about STLC?** Software Testing Life Cycle (STLC) defines the stages of testing in software development, including requirement analysis, test planning, test case design, test execution, and defect tracking.
26. **In software testing, what does a fault mean?** A fault is a defect or imperfection in the software that causes incorrect behavior.
27. **How do severity and priority relate to each other?** Severity refers to the impact of the defect on the system, while priority refers to the urgency with which the defect should be fixed.
28. **Can you describe the defect life cycle and the different stages involved in defect management?** The defect life cycle includes stages like New, Assigned, Open, Fixed, Retested, and Closed. It tracks the defect from discovery to resolution.
29. **What techniques do you use to test database views and stored procedures using SQL queries? How do you ensure their correctness and efficiency?** Techniques include boundary testing, data validation, and performance testing using SQL queries. Ensure correctness by checking the output against expected results and efficiency by analyzing query performance.
30. **Describe the process you follow for creating and executing test cases in a manual testing environment.** Create test cases based on requirements, define inputs, expected outputs, and execute tests step-by-step while documenting results and defects.
31. **What do you mean by defect detection percentage in software testing?** Defect detection percentage is a metric that indicates the proportion of defects identified during testing compared to total defects (including those found in production).
32. **What is the average age of a defect in software testing?** The average age of a defect is the time between when the defect is identified and when it is resolved.
33. **On the basis of which factors would you consider choosing automated testing over manual testing?** Automated testing is chosen when testing is repetitive, time-consuming, requires high accuracy, or needs to be performed across multiple environments.
34. **Is there any difference between bug leakage and bug release?** Bug leakage refers to defects found after release, while bug release refers to the intentional release of a bug for future resolution or in cases where it’s deemed low priority.
35. **What is meant by latent defect?** A latent defect is a defect that remains undetected until later stages of the software’s life cycle, typically after release.