ISTQB-FL

ADVANCED TOP INTERVIEW QUESTIONS AND ANSWERS

Part- 3

Prepare By :

Mariam Elnshar

Software Testing Engineer

Mariam.elnshar97@gmail.com

01201956860

1. What is Exploratory Testing?

Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used. The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session so that a report can be produced afterward.

2. What is “use case testing”?

In order to identify and execute the functional requirement of an application from start to finish “use case” is used and the techniques used to do this is known as “Use Case Testing.”

3. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

SDLC deals with development/coding of the software while STLC deales with validation and verification of the software

4. What is traceability matrix?

The relationship between test cases and requirements is shown with the help of a document. This document is known as a traceability matrix.

5. What is Equivalence partitioning testing?

Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived. By this testing method, it reduces the time required for software testing.

6. What is white box testing and list the types of white box testing?

White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage, etc.) of a component or system. It is also known as Code-Based testing or Structural testing. Different types of white box testing are

1. Statement Coverage
2. Decision Coverage

7. In white box testing, what do you verify?

In white box testing following steps are verified.

1. Verify the security holes in the code
2. Verify the incomplete or broken paths in the code
3. Verify the flow of structure according to the document specification
4. Verify the expected outputs
5. Verify all conditional loops in the code to check the complete functionality of the application
6. Verify the line by line coding and cover 100% testing

8. What is black box testing? What are the different black box testing techniques?

Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program. This testing is usually done to check the functionality of an application. The different black box testing techniques are

1. Equivalence Partitioning
2. Boundary value analysis
3. Cause-effect graphing

9. What is the difference between static and dynamic testing?

Static testing: During Static testing method, the code is not executed, and it is performed using the software documentation.

Dynamic testing: To perform this testing the code is required to be in an executable form.

10. What are verification and validation?

Verification is a process of evaluating software at the development phase. It helps you to decide whether the product of a given application satisfies the specified requirements. Validation is the process of evaluating software at the after the development process and to check whether it meets the customer requirements.

11. What are the different test levels?

There are four test levels

1. Unit/component/program/module testing
2. Integration testing
3. System testing
4. Acceptance testing

12. What is Integration testing?

Integration testing is a level of software testing process, where individual units of an application are combined and tested. It is usually performed after unit and functional testing.

13. What Test Plans consists of?

Test design, scope, test strategies, approach are various details that Test plan document consists of.

1. Test case identifier
2. Scope
3. Features to be tested
4. Features not to be tested
5. Test strategy & Test approach
6. Test deliverables
7. Responsibilities
8. Staffing and training
9. Risk and Contingencies

14. What is the difference between UAT (User Acceptance Testing) and System testing?

System Testing: System testing is finding defects when the system undergoes testing as a whole; it is also known as end-to-end testing. In such type of testing, the application suffers from beginning till the end.

UAT: User Acceptance Testing (UAT) involves running a product through a series of specific tests which determines whether the product will meet the needs of its users.

15. Mention the difference between Data Driven Testing and Retesting?

Retesting: It is a process of checking bugs that are actioned by the development team to verify that they are fixed.

Data Driven Testing (DDT): In data driven testing process, the application is tested with multiple test data. The application is tested with a different set of values.

16. What are the valuable steps to resolve issues while testing?

- Record: Log and handle any problems which have happened
- Report: Report the issues to higher level manager
- Control: Define the issue management process

17. What is the difference between test scenarios, test cases, and test script?

Difference between test scenarios and test cases is that

Test Scenarios: A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.

Test Cases: It is a document that contains the steps that have to be executed; it has been planned earlier.

Test Script: It is written in a programming language and it's a short program used to test part of the functionality of the software system. In other words a written set of steps that should be performed manually.

18. What is Latent defect?

Latent defect: This defect is an existing defect in the system which does not cause any failure as the exact set of conditions has never been met

19. What are the two parameters which can be useful to know the quality of test execution?

To know the quality of test execution, we can use two parameters

- Defect reject ratio
- Defect leakage ratio

Parameters for quality of test execution

20. What is the function of the software testing tool “phantom”?

Phantom is a freeware and is used for windows GUI automation scripting language. It allows us to take control of windows and functions automatically. It can simulate any combination of keystrokes and mouse clicks as well as menus, lists and more.

21. Explain what Test Deliverables is?

Test Deliverables are a set of documents, tools and other components that have to be developed and maintained in support of testing.

There are different test deliverables at every phase of the software development lifecycle

- Before Testing
- During Testing
- After the Testing

22. What is mutation testing?

Mutation testing is a technique to identify if a set of test data or test case is useful by intentionally introducing various code changes (bugs) and retesting with original test data/ cases to determine if the bugs are detected.

23. What all things you should consider before selecting automation tools for the AUT?

- Technical Feasibility
- Complexity level
- Application stability
- Test data
- Application size
- Re-usability of automated scripts
- Execution across environment

24. How will you conduct Risk Analysis?

For the risk analysis following steps need to be implemented

1. Finding the score of the risk
2. Making a profile for the risk
3. Changing the risk properties
4. Deploy the resources of that test risk
5. Making a database of risk

25. What are the categories of debugging?

Categories for debugging

1. Brute force debugging
2. Backtracking
3. Cause elimination
4. Program Slicing
5. Fault tree analysis

26. What is fault masking explain with example?

When the presence of one defect hides the presence of another defect in the system, it is known as fault masking.

Example: If the “Negative Value” cause a firing of unhandled system exception, the developer will prevent the negative values input. This will resolve the issue and hide the defect of unhandled exception firing.

27. Explain what Test Plan is? What is the information that should be covered in Test Plan?

A test plan can be defined as a document describing the scope, approach, resources, and schedule of testing activities and a test plan should cover the following details.

- Test Strategy
- Test Objective
- Exit/Suspension Criteria
- Resource Planning
- Test Deliverables

28. How can you eliminate the product risk in your project?

It helps you to eliminate product risk in your project, and there is a simple yet crucial step that can reduce the product risk in your project.

- Investigate the specification documents
- Have discussions about the project with all stakeholders including the developer
- As a real user walk around the website

29. What is the common risk that leads to project failure?

The common risk that leads to a project failure are

- Not having enough human resource
- Testing Environment may not be set up properly
- Limited Budget
- Time Limitations

30. On what basis you can arrive at an estimation for your project?

To estimate your project, you have to consider the following points

- Divide the whole project into the smallest tasks
- Allocate each task to team members
- Estimate the effort required to complete each task

31. Explain how you would allocate a task to team members?

Task	Member
• Analyze software requirement specification	• All the members
• Create the test specification	• Tester/Test Analyst
• Build up the test environment	• Test administrator
• Execute the test cases	• Tester, a Test administrator
• Report defects	• Tester

32. Explain what is testing type and what are the commonly used testing type?

To get an expected test outcome, a standard procedure is followed which is referred to as Testing Type.

Commonly used testing types are

- Unit Testing: Test the smallest code of an application
- API Testing: Testing API created for the application
- Integration Testing: Individual software modules are combined and tested
- System Testing: Complete testing of the system
- Install/Uninstall Testing: Testing done from the point of client/customer view
- Agile Testing: Testing through Agile technique

33. While monitoring your project what all things you have to consider?

The things that have to be taken in considerations are

- Is your project on schedule
- Are you over budget
- Are you working towards the same career goal
- Have you got enough resources
- Are there any warning signs of impending problems
- Is there any pressure from management to complete the project sooner

34. What are the common mistakes which create issues?

- Matching resources to wrong projects
- Test manager lack of skills
- Not listening to others
- Poor Scheduling
- Underestimating
- Ignoring the small problems
- Not following the process

35. What does a typical test report contain? What are the benefits of test reports?

A test report contains the following things:

- Project Information
- Test Objective
- Test Summary
- Defect

The benefits of test reports are:

- Current status of project and quality of product are informed
- If required, stakeholder and customer can take corrective action
- A final document helps to decide whether the product is ready for release

36. What is test management review and why it is important?

Management review is also referred to as Software Quality Assurance or SQA. SQA focusses more on the software process rather than the software work products. It is a set of activities designed to make sure that the project manager follows the standard process. SQA helps test manager to benchmark the project against the set standards.

37. What are the best practices for software quality assurance?

The best practices for an effective SQA implementation is

- Continuous Improvement
- Documentation
- Tool Usage
- Metrics
- Responsibility by team members
- Experienced SQA auditors

38. When is RTM (Requirement Traceability Matrix) prepared?

RTM is prepared before test case designing. Requirements should be traceable from review activities.

39. What is the difference between Test matrix and Traceability matrix?

Test Matrix: Test matrix is used to capture actual quality, effort, the plan, resources and time required to capture all phases of software testing

Traceability Matrix: Mapping between test cases and customer requirements is known as Traceability Matrix

40. In manual testing what are stubs and drivers?

Both stubs and drivers are part of incremental testing. In incremental testing, there are two approaches namely bottom-up and top-down approach. Drivers are used in bottom-up testing and stub is used for a top-down approach. In order to test the main module, the stub is used, which is a dummy code or program.

41. What is the step you would follow once you find the defect?

Once a defect is found you would follow the step

a) Recreate the defect

b) Attach the screenshot

c) Log the defect

42. Explain what is “Test Plan Driven” or “Key Word Driven” method of testing?

This technique uses the actual test case document developed by testers using a spreadsheet containing special “key Words”. The key words control the processing.

43. What is the DFD (Data Flow Diagram)?

When a “flow of data” through an information system is graphically represented, then it is known as Data Flow Diagram. It is also used for the visualization of data processing.

44. Explain what LCSAJ is?

LCSAJ stands for 'linear code sequence and jump.' It consists of the following three items

a) Start of the linear sequence of executable statements

b) End of the linear sequence

c) The target line to which control flow is transferred at the end of the linear sequence

45. Explain what N+1 testing is?

The variation of regression testing is represented as N+1. In this technique, the testing is performed in multiple cycles in which errors found in test cycle 'N' are resolved and re-tested in test cycle N+1. The cycle is repeated unless there are no errors found.

46. What is Fuzz testing and when it is used?

Fuzz testing is used to detect security loopholes and coding errors in software. In this technique, random data is added to the system in an attempt to crash the system. If vulnerability persists, a tool called fuzz tester is used to determine potential causes. This technique is more useful for bigger projects but only detects a major fault.

47. Mention what the main advantages of statement coverage metric of software testing are?

The benefit of statement coverage metric is that

a) It does not require processing source code and can be applied directly to object code

b) Bugs are distributed evenly through the code, due to which percentage of executable statements covered reflects the percentage of faults discovered

48. How to generate test cases for “replace a string” method?

a) If characters in new string > characters in the previous string. None of the characters should get truncated

b) If characters in new string< characters in the previous string. Junk characters should not be added

c) Spaces after and before the string should not be deleted

d) String should be replaced only for the first occurrence of the string

49. How will you handle a conflict amongst your team members?

- I will talk individually to each person and note their concerns
- I will find a solution to the common problems raised by team members
- I will hold a team meeting, reveal the solution and ask people to co-operate

50. Mention what are the categories of defects?

Mainly there are three defect categories

- **Wrong**: When a requirement is implemented incorrectly
- **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
- **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements