

1. What is Exploratory Testing?

- Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution. It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests.

2. What is traceability matrix?

- A traceability matrix is a document that details the technical requirements for a given test scenario and its current state. It helps the testing team understand the level of testing that is done for a given product. The traceability process itself is used to review the test cases that were defined for any requirement.

3. What is Boundary value testing?

- Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.

4. What is Equivalence partitioning testing?

- Equivalence partitioning is a technique that divides the input domain of a system into partitions or classes that are expected to produce the same output or behavior.

5. What is Integration testing?

- Integration testing is known as the second level of the software testing process, following unit testing. Integration testing involves checking individual components or units of a software project to expose defects and problems to verify that they work together as designed.

6. What determines the level of risk?

- Determining the level of risk usually involves trying to assess not only the likelihood of an identified risk from actually occurring, but also the potential magnitude the consequences this risk could have on an organisation and its stakeholder, should it occur.

7. What is Alpha testing?

- Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly. It is typically performed by internal employees and conducted in a lab/stage environment. An alpha test ensures the product really works and does everything it's supposed to do.

8. What is beta testing?

- Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release. Beta testing is the final round of testing before releasing a product to a wide audience.

9. What is component testing?

- Component testing is a form of closed-box testing, meaning that the test evaluates the behavior of the program without considering the details of the underlying code. Component testing is done on the section of code in its entirety, after the development has been completed.

10. What is functional system testing?

- Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

11. What is Non-Functional Testing?

- Non-functional testing is almost as critical as functional testing, and it has an effect on customer satisfaction.

12. What is GUI Testing?

- GUI testing refers to the validating UI functions or features of an application that are visible to the users, and they should comply with business requirements.

13. What is Adhoc testing?

- Adhoc testing is a type of software testing which is performed informally and randomly after the formal testing is completed to find out any loophole in the system. For this reason, it is also known as Random testing or Monkey testing.

14. What is load testing?

- Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users.

15. What is stress Testing?

- Stress testing is defined as a type of software testing that verifies the stability and reliability of the system. This test particularly determines the system on its robustness and error handling under extremely heavy load conditions.

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

- White box testing is a form of application testing that provides the tester with complete knowledge of the application being tested, including access to source code and design documents.
 - ❖ list the types of white box testing
 - Path Testing is a white-box testing approach based on a program's control structure.
 - Loop Testing
 - Conditional Testing
 - Unit Testing
 - Mutation Testing
 - Integration Testing
 - Penetration Testing
 - Testing based on Memory Perspective
 - Test Performance of the Program

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

- Black box testing, a form of testing that is performed with no knowledge of a system's internals, can be carried out to evaluate the functionality, security, performance, and other aspects of an application. Dynamic code analysis is an example of automated black box security testing.
 - ❖ There are four main black box testing techniques:
 - equivalence partitioning
 - boundary value analysis
 - decision table testing
 - state transition testing

18. Mention what are the categories of defects?

- Arithmetic Defects.
- Logical Defects.
- Syntax Defects.
- Multithreading Defects.
- Interface Defects.
- Performance Defects.

19. Mention what bigbang testing is?

- Big-bang integration testing is a type of integration testing that combines all the modules or components of a system into a single unit and tests them as a whole.

20. What is the purpose of exit criteria?

- Exit criteria is an important document prepared by the QA team to adhere to the imposed deadlines and allocated budget. This document specifies the conditions and requirements that are required to be achieved or fulfilled before the end of software testing process.

21. When should "Regression Testing" be performed?

- Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

22. what is 7 key principales? Explain in detail.

- 1) Exhaustive testing is not possible

Yes! Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.

And the million dollar question is, how do you determine this risk?

To answer this let's do an exercise

In your opinion, Which operation is most likely to cause your Operating system to fail?

I am sure most of you would have guessed, Opening 10 different application all at the same time.

So if you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next principle Defect Clustering

- 2) Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.

By experience, you can identify such risky modules. But this approach has its own problems

If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

- 3) Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide. Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

To overcome this, the test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.

Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective. But even after all this sweat & hard work in testing, you can never claim your product is bug-free. To drive home this point, let's see this video of the public launch of Windows 98

You think a company like MICROSOFT would not have tested their OS thoroughly & would risk their reputation just to see their OS crashing during its public launch!

4) Testing shows a presence of defects

Hence, testing principle states that – Testing talks about the presence of defects and don't talk about the absence of defects. i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness. But what if, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

This leads us to our next principle, which states that- Absence of Error

5) Absence of Error – fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.

To solve this problem, the next principle of testing states that Early Testing

6) Early Testing

Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

7) Testing is context dependent

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed

software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

23. Difference between QA vs QC Tester

- While QA testing focuses on providing assurance that quality requested will be achieved, QC testing focuses on fulfilling the quality requested. QA focuses on preventing defects while QC focuses on identifying the defect.

24. Difference between smoke and sanity

- Smoke testing is executed at the initial stage of SDLC, to check the core functionalities of an application. Whereas Sanity & Regression testing are done at the final stage of SDLC, to check the main functionalities of an application.

25. Difference between verification and validation

- Validation is the process of checking whether the specification captures the customer's requirements, while verification is the process of checking that the software meets specifications. Verification includes all the activities associated with the producing high quality software.

26. Explain types of Performance testing.

- Performance testing is a testing measure that evaluates the speed, responsiveness and stability of a computer, network, software program or device under a workload. Organizations will run performance tests to identify performance-related bottlenecks.

27. What is error, Defect, Bug and Failure?

- Error
- An error is a mistake made by a human that leads to a discrepancy between the actual and the expected result.
- Defect
- A defect is a problem in the functioning of a software system during testing. ISTQB defines a defect as "A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g., an incorrect statement or data definition."
- Bug
- A bug is a flaw in a software system that causes the system to behave in an unintended manner.

- Failure
- A failure is the inability of a software system to perform its operations within the specified performance benchmark. As per ISTQB, “a defect, if encountered during execution, may cause a failure of the component or system”.

28. Difference between Priority and Severity?

- Priority is a term that defines how fast we need to fix a defect. Severity is basically a parameter that denotes the total impact of a given defect on any software. Priority is basically a parameter that decides the order in which we should fix the defects. Severity relates to the standards of quality.

29. what is bug cycle?

- The bug life cycle in testing refers to a cycle of defects in which it goes through different states throughout its life. The life cycle begins with a new defect discovered by a tester while testing the application. It continues until the tester discovers a specific solution and closes the bug, so it does not reoccur.

30. Difference between functional and non-functional testing

- Functional testing is the testing in which the basic functionality, operations and actions of the application/software on the basis of requirements provided is done. Non-Functional testing is the testing in which the performance or usability and behavior of software/application is done under different circumstances.