1. What is Software Testing?

According to ANSI/IEEE 1059 standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

2. What are Quality Assurance and Quality Control is?

Quality Assurance: Quality Assurance involves in process-oriented activities. It ensures the prevention of defects in the process used to make Software Application. So the defects don't arise when the Software Application is being developed.

Quality Control: Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

3. What is Verification in software testing?

Verification is the process, to ensure that whether we are building the product right i.e., to verify the requirements which we have and to verify whether we are developing the product accordingly or not. Activities involved here are Inspections, Reviews, Walk-throughs.

4. What is Validation in software testing?

Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not. Activities involved in this is Testing the software application.

5. What is Static Testing?

Static Testing involves in reviewing the documents to identify the defects in the early stages of SDLC.

6. What is Dynamic Testing?

Dynamic testing involves in the execution of code. It validates the output with the expected outcome.

7. What is White Box Testing?

White Box Testing is also called as Glass Box, Clear Box, and Structural Testing. It is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually was done at the unit level.

8. What is Black Box Testing?

Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

9. What is Grey Box Testing?

Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents.
This helps to create better test cases in this process.

10. What is Positive and Negative Testing?

Positive Testing: It is to determine what system supposed to do. It helps to check whether the application is justifying the requirements or not.

Negative Testing: It is to determine what system not supposed to do. It helps to find the defects from the software.

11. What is Test Strategy?

Test Strategy is a high-level document (static document) and usually developed by project manager. It is a document which captures the approach on how we go about testing the product and achieve the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as a base.

12. What is Test Plan and contents available in a Test Plan?

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product. Test Plan document is derived from the Product Description, SRS, or Use Case documents for all future activities of the project. It is usually prepared by the Test Lead or Test Manager.

1. Test plan identifier
2. References
3. Introduction
4. Test items (functions)
5. Software risk issues
6. Features to be tested
7. Features not to be tested
8. Approach

9. Items pass/fail criteria

10. Suspension criteria and resolution requirements

11. Test deliverables

12. Remaining test tasks

13. Environmental needs

14. Staff and training needs

15. Responsibility

16. Schedule

17. Plan risks and contingencies

18. Approvals

19. Glossaries

13. What is Test Suite?

Test Suite is a collection of test cases. The test cases which are intended to test an application.

14. What is Test Scenario?

Test Scenario gives the idea of what we have to test. Test Scenario is like a high level test case.

15. What is Test Case?

Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post conditions and actual results.

16. What is Test Bed?

An environment configured for testing. Test bed consists of hardware, software, network configuration, an application under test, other related software.

17. What is Test Environment?

Test Environment is the combination of hardware and software on which Test Team performs testing.

Example:

Application Type: web application
OS: windows
Web Server: IIS

Web Page Design: Dot Net
Client Side Validation: JavaScript
Server Side Scripting: ASP Dot Net
Database: MS SQL Server Browser:
IE/FireFox/Chrome

18. What is Test Data?

Test data is the data that is used by the testers to run the test cases. Whilst running the test cases, testers need to enter some input data. To do so, testers prepare test data. It can be prepared manually and also by using tools.
For example, To test a basic login functionality having a user id, password fields. We need to enter some data in the user id and password fields. So we need to collect some test data.

19. What is Test Harness?

A test harness is the collection of software and test data configured to test a program unit by running it under varying conditions which involves monitoring the output with expected output.

20. What is Test Closure?

Test Closure is the note prepared before test team formally completes the testing process. This note contains the total no. of test cases, total no. of test cases executed, total no. of defects found, total no. of defects fixed, total no. of bugs not fixed, total no of bugs rejected etc.,

21. What are the tasks of Test Closure activities in Software Testing?

Test Closure activities fall into four major groups.

Test Completion Check: To ensure all tests should be either run or deliberately skipped and all known defects should be either fixed, deferred for a future release or accepted as a permanent restriction.

Test Artifacts handover: Tests and test environments should be handed over to those responsible for maintenance testing. Known defects accepted or deferred should be documented and communicated to those who will use and support the use of the system.

Lessons learned: Analyzing lessons learned to determine changes needed for future releases and projects. In retrospective meetings, plans are established to ensure that good practices can be repeated and poor practices are not repeated

Archiving results, logs, reports, and other documents and work products in the CMS (configuration management system).

22. What is test coverage?

Test coverage helps in measuring the amount of testing performed by a set of tests. Test coverage can be done on both functional and non-functional activities. It assists testers to create tests that cover areas which are missing.

23. What is Code coverage?

Code coverage is different from Test coverage. Code coverage is about unit testing practices that must target all areas of the code at least once. It is usually done by developers or unit testers.

Refer Test Metrics.

24. List out Test Deliverables?

1. Test Strategy
2. Test Plan
3. Effort Estimation Report
4. Test Scenarios
5. Test Cases/Scripts
6. Test Data
7. Requirement Traceability Matrix (RTM)
8. Defect Report/Bug Report
9. Test Execution Report
10. Graphs and Metrics
11. Test summary report
12. Test incident report
13. Test closure report
14. Release Note
15. Installation/configuration guide
16. User guide
17. Test status report
18. Weekly status report (Project manager to client)

25. What is Unit Testing?

Unit Testing is also called as Module Testing or Component Testing. It is done to check whether the individual unit or module of the source code is working properly. It is done by the developers in the developer's environment.

26. What is Integration Testing?

Integration Testing is the process of testing the interface between the two software units. Integration testing is done by three ways. Big Bang Approach, Top-Down Approach, Bottom-Up Approach

27. What is System Testing?

Testing the fully integrated application to evaluate the system's compliance with its specified requirements is called System Testing AKA End to End testing. Verifying the completed system to ensure that the application works as intended or not.

28. What is Big Bang Approach?

Combining all the modules once and verifying the functionality after completion of individual module testing.

Top down and bottom up are carried out by using dummy modules known as Stubs and Drivers. These Stubs and Drivers are used to stand-in for missing components to simulate data communication between modules.

29. What is Top-Down Approach?

Testing takes place from top to bottom. High-level modules are tested first and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended. Stubs are used as a temporary module if a module is not ready for integration testing.

30. What is Bottom-Up Approach?

It is a reciprocate of the Top-Down Approach. Testing takes place from bottom to up. Lowest level modules are tested first and then high-level modules and finally integrating the high-level modules to a low level to ensure the system is working as intended. Drivers are used as a temporary module for integration testing.

31. What is End-To-End Testing?

Refer System Testing.

32. What is Functional Testing?

In simple words, what the system actually does is functional testing. To verify that each function of the software application behaves as specified in the requirement document. Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not. It falls within the scope of black box testing and the testers need not concern about the source code of the application.

33. What is Non-Functional Testing?

In simple words, how well the system performs is non-functionality testing. Nonfunctional testing refers to various aspects of the software such as performance, load, stress, scalability, security, compatibility etc., Main focus is to improve the user experience on how fast the system responds to a request.

34. What is Acceptance Testing?

It is also known as pre-production testing. This is done by the end users along with the testers to validate the functionality of the application. After successful acceptance testing. Formal testing conducted to determine whether an application is developed as per the requirement. It allows the customer to accept or reject the application. Types of acceptance testing are Alpha, Beta & Gamma.

35. What is Alpha Testing?

Alpha testing is done by the in-house developers (who developed the software) and testers. Sometimes alpha testing is done by the client or outsourcing team with the presence of developers or testers.

36. What is Beta Testing?

Beta testing is done by a limited number of end users before delivery. Usually, it is done in the client place.

37. What is Gamma Testing?

Gamma testing is done when the software is ready for release with specified requirements. It is done at the client place. It is done directly by skipping all the in house testing activities.

38. What is Smoke Testing?

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as "Day 0" check. It is done at the "build level". It helps not to

waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet.

39. What is Sanity Testing?

Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of Regression testing. It is done at the "release level". At times due to release time constraints rigorous regression testing can't be done to the build, sanity testing does that part by checking main functionalities.

40. What is Retesting?

To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build. Say, Build 1.0 was released. Test team found some defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

41. What is Regression Testing?

Repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.

Usually, we do regression testing in the following cases:
1. New functionalities are added to the application
2. Change Requirement (In organizations, we call it as CR)
3. Defect Fixing
4. Performance Issue Fix
5. Environment change (E.g., Updating the DB from MySQL to Oracle)

42. What is GUI Testing?

Graphical User Interface Testing is to test the interface between the application and the end user.

43. What is Recovery Testing?

Recovery testing is performed in order to determine how quickly the system can recover after the system crash or hardware failure. It comes under the type of non-functional testing.

44. What is Globalization Testing?
Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.

45. What is Internationalization Testing (I18N Testing)?

Refer Globalization Testing.

46. What is Localization Testing (L10N Testing)?

Localization is a process of adapting globalization software for a specific region or language by adding local specific components.

47. What is Installation Testing?
It is to check whether the application is successfully installed and it is working as expected after installation.

48. What is Formal Testing?
It is a process where the testers test the application by having pre-planned procedures and proper documentation.

49. What is Risk Based Testing?

Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.

50. What is Compatibility Testing?
It is to deploy and check whether the application is working as expected in a different combination of environmental components.

51. What is Exploratory Testing?
Usually, this process will be carried out by domain experts. They perform testing just by exploring the functionalities of the application without having the knowledge of the requirements.

52. What is Monkey Testing?

Perform abnormal action on the application deliberately in order to verify the stability of the application.

53. What is Usability Testing?

To verify whether the application is user-friendly or not and was comfortably used by an end user or not. The main focus in this testing is to check whether the end user can understand and operate the application easily or not. An application should be self-exploratory and must not require training to operate it.

54. What is Security Testing?

Security testing is a process to determine whether the system protects data and maintains functionality as intended.

55. What is Soak Testing?

Running a system at high load for a prolonged period of time to identify the performance problems is called Soak Testing.

56. What is Endurance Testing?

Endurance testing is a non-functional testing type. It is also known as Soak Testing. Refer Soak testing.

57. What is Performance Testing?

This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the project or product.

58. What is Load Testing?

It is to verify that the system/application can handle the expected number of transactions and to verify the system/application behavior under both normal and peak load conditions.

59. What is Volume Testing?

It is to verify that the system/application can handle a large amount of data

60. What is Stress Testing?

It is to verify the behavior of the system once the load increases more than its design expectations.

61. What is Scalability Testing?

Scalability testing is a type of non-functional testing. It is to determine how the application under test scales with increasing workload.

62. What is Concurrency Testing?

Concurrency testing means accessing the application at the same time by multiple users to ensure the stability of the system. This is mainly used to identify deadlock issues.

63. What is Fuzz Testing?

Fuzz testing is used to identify coding errors and security loopholes in an application. By inputting massive amount of random data to the system in an attempt to make it crash to identify if anything breaks in the application.

64. What is Adhoc Testing?

Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Adhoc testing, testers randomly test the application without following any documents and test design techniques. This testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without any test cases or any business requirement document.

65. What is Interface Testing?

Interface testing is performed to evaluate whether two intended modules pass data and communicate correctly to one another.

66. What is Reliability Testing?
Perform testing on the application continuously for long period of time in order to verify the stability of the application

67. What is Bucket Testing?

Bucket testing is a method to compare two versions of an application against each other to determine which one performs better.

68. What is A/B Testing?

Refer Bucket Testing.

69. What is Split Testing?

Refer Bucket Testing.
70. What are the principles of Software Testing?

1. Testing shows presence of defects
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide Paradox
6. Testing is context depending

7. Absence of error fallacy

71. What is Exhaustive Testing?

Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing.

72. What is Early Testing?

Defects detected in early phases of SDLC are less expensive to fix. So conducting early testing reduces the cost of fixing defects.

73. What is Defect clustering?

Defect clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures.

74. What is Pesticide Paradox?

Pesticide Paradox in software testing is the process of repeating the same test cases, again and again, eventually, the same test cases will no longer find new bugs. So to overcome this Pesticide Paradox, it is necessary to review the test cases regularly and add or update them to find more defects.

75. What is Defect Cascading in Software Testing?

Defect cascading in Software testing means triggering of other defects in an application. When a defect is not identified or goes unnoticed while testing, it invokes other defects. It leads to multiple defects in the later stages and results in an increase in a number of defects in the application.

For example, if there is a defect in an accounting system related to negative taxation then the negative taxation defect affects the ledger which in turn affects other reports such as Balance Sheet, Profit & Loss etc.,

76. What is Walk Through?

A walkthrough is an informal meeting conducts to learn, gain understanding, and find defects. The author leads the meeting and clarifies the queries raised by the peers in the meeting.

77. What is Inspection?

Inspection is a formal meeting lead by a trained moderator, certainly not by the author. The document under inspection is prepared and checked thoroughly by the reviewers before the

meeting. In the inspection meeting, the defects found are logged and shared with the author for appropriate actions. Post inspection, a formal follow-up process is used to ensure a timely and corrective action.

78. Who are all involved in an inspection meeting?

Author, Moderator, Reviewer(s), Scribe/Recorder and Manager.

79. What is a Defect?

The variation between the actual results and expected results is known as a defect. If a developer finds an issue and corrects it by himself in the development phase then it's called a defect.

80. What is a Bug?

If testers find any mismatch in the application/system in testing phase then they call it as Bug.

81. What is an Error?

We can't compile or run a program due to a coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an error.

82. What is a Failure?

Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as a failure.

83. What is Bug Severity?

Bug/Defect severity can be defined as the impact of the bug on customer's business. It can be Critical, Major or Minor. In simple words, how much effect will be there on the system because of a particular defect.

84. What is Bug Priority?

Defect priority can be defined as how soon the defect should be fixed. It gives the order in which a defect should be resolved. Developers decide which defect they should take up next based on the priority. It can be High, Medium or Low. Most of the times the priority status is set based on the customer requirement.

85. Tell some examples of Bug Severity and Bug Priority?

High Priority & High Severity: Submit button is not working on a login page and customers are unable to login to the application

Low Priority & High Severity: Crash in some functionality which is going to deliver after couple of releases

High Priority & Low Severity: Spelling mistake of a company name on the homepage

Low Priority & Low Severity: FAQ page takes a long time to load

86. What is a Critical Bug?

A critical bug is a show stopper which means a large piece of functionality or major system component is completely broken and there is no workaround to move further.
For example, Due to a bug in one module, we cannot test the other modules because that blocker bug has blocked other modules. Bugs which affects the customers business are considered as critical.

Example:

1. "Sign In" button is not working on Gmail App and Gmail users are blocked tologin to their accounts.
2. An error message pops up when a customer clicks on transfer money button ina Banking website.

87. What is the difference between a Standalone application, Client-Server application and Web application?

Standalone application:

Standalone applications follow one-tier architecture. Presentation, Business, and Database layer are in one system for a single user.

Client-Server Application:

Client-server applications follow two-tier architecture. Presentation and Business layer are in a client system and Database layer on another server. It works majorly in Intranet.

Web Application:

Web server applications follow three-tier or n-tier architecture. The presentation layer is in a client system, a Business layer is in an application server and Database layer is in a Database server. It works both in Intranet and Internet.

88. What is Bug Life Cycle?

Bug life cycle is also known as **Defect life cycle**. In Software Development process, the bug has a life cycle. The bug should go through the life cycle to be closed. Bug life cycle varies depends upon the tools (QC, JIRA etc.,) used and the process followed in the organization.

89. What is Bug Leakage?

A bug which is actually missed by the testing team while testing and the build was released to the Production. If now that bug (which was missed by the testing team) was found by the end user or customer then we call it as Bug Leakage.

90. What is Bug Release?

Releasing the software to the Production with the known bugs then we call it as Bug Release. These known bugs should be included in the release note.

91. What is Defect Age?

Defect age can be defined as the time interval between date of defect detection and date of defect closure.

Defect Age = Date of defect closure – Date of defect detection

Assume, a tester found a bug and reported it on 1 Jan 2016 and it was successfully fixed on 5 Jan 2016. So the defect age is 5 days.

92. What is Error Seeding?

Error seeding is a process of adding known errors intendedly in a program to identify the rate of error detection. It helps in the process of estimating the tester skills of finding bugs and also to know the ability of the application (how well the application is working when it has errors.)

93. What is Showstopper Defect?

A showstopper defect is a defect which won't allow a user to move further in the application. It's almost like a crash.

Assume that login button is not working. Even though you have a valid username and valid password, you could not move further because the login button is not functioning.

94. What is HotFix?

A bug which needs to handle as a high priority bug and fix it immediately.

95. What is Boundary Value Analysis?

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each 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. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition. A boundary value for a valid partition is a valid boundary value. Similarly, a boundary value for an invalid partition is an invalid boundary value.

96. What is Equivalence Class Partition?

Equivalence Partitioning is also known as Equivalence Class Partitioning. In equivalence partitioning, inputs to the software or system are divided into groups that are expected to exhibit similar behavior, so they are likely to be proposed in the same way. Hence selecting one input from each group to design the test cases.

97. What is Decision Table testing?

Decision Table is aka Cause-Effect Table. This test technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In Decision table technique, we deal with combinations of inputs. To identify the test cases with decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs.

98. What is State Transition?

Using state transition testing, we pick test cases from an application where we need to test different system transitions. We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state.

99. What is an entry criteria?

The prerequisites that must be achieved before commencing the testing process.

100. What is an exit criteria?

The conditions that must be met before testing should be concluded.

101. What is SDLC?

Software Development Life Cycle (SDLC) aims to produce a high-quality system that meets or exceeds customer expectations, works effectively and efficiently in the current and planned information technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.

102. What are the different available models of SDLC?

1. Waterfall
2. Spiral
3. V Model
4. Prototype
5. Agile

103. Can you do System testing at any stage of SDLC?

We can do System Testing only when all the units are in place and working properly. It can only be done before User Acceptance Testing (UAT).

104. What is the procedure of manual testing?

Manual testing is crucial for testing software applications more thoroughly. The procedure of manual testing comprises of the following.
1. Planning and Control
2. Analysis and Design
3. Implementation and Execution
4. Evaluating and Reporting
5. Test Closure activities

Refer Software Development Life Cycle (SDLC) & Software Testing Life Cycle (STLC)

105. What is STLC?

STLC (Software Testing Life Cycle) identifies what test activities to carry out and when to accomplish those test activities. Even though testing differs between Organizations, there is a testing life cycle.

106. What is RTM?

Requirements Traceability Matrix (RTM) is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled. Requirement Traceability Matrix AKA Traceability Matrix or Cross Reference Matrix.

107. What is Test Metrics?

Software test metrics is to monitor and control process and product. It helps to drive the project towards our planned goals without deviation. Metrics answer different questions. It's important to decide what questions you want answers to.

108. When to stop testing? (Or) How do you decide when you have tested enough?

There are many factors involved in the real-time projects to decide when to stop testing.

1. Testing deadlines or release deadlines
2. By reaching the decided pass percentage of test cases
3. The risk in the project is under acceptable limit
4. All the high priority bugs, blockers are fixed
5. When acceptance criteria is met