ECHLEON INSTITUTE OF TECHNOLOGY
QUESTION BANK

Ques.1. What do you mean by Software Testing?

Ans. Software testing is the process of evaluating a system to check if it satisfies its business requirements. It measures the overall quality of the system in terms of attributes. Like – correctness, completeness, usability, performance, etc.

Basically, it is used for ensuring the quality of software to the stakeholders of the application.

Ques.2. Why is testing required?

Ans. We need software testing for the following reasons-

1. Testing provides an assurance to the stakeholders that the product works as intended.

2. Avoidable defects leaked to the end-user/customer without proper testing adds a bad reputation to the development company.

3. Defects detected earlier phase of SDLC results in lesser cost and resource utilization of correction.

4. Saves development time by detecting issues in an earlier phase of development.

5. The testing team adds another dimension to the software development by providing a different viewpoint to the product development process.

Ques.3. When should we stop testing?

Ans. Testing (both manual and automated) can be stopped when one or more of the following conditions are met-

1. **After test case execution** – The testing phase can be stopped when one complete cycle of test cases is executed after the last known bug fix with the agreed-upon value of pass-percentage

2. **Once the testing deadline is met** – Testing can be stopped after deadlines get met with no high-priority issues left in the system.

3. Based on Mean Time Between Failure (MTBF) – MTBF is the time interval between two inherent failures. Based on stakeholders' decisions, if the MTBF is quite large, one can stop the testing phase.

4. Based on code coverage value – The testing phase can be stopped when the automated code coverage reaches a specific threshold value with sufficient pass percentage and no critical bug.

Ques.4. What is Quality Assurance and what are the different activities involved in Quality assurance?

Ans. Quality assurance is a process-driven approach that checks if the process of developing the product is correct and conforming to all the standards. It is considered a preventive measure. This is because it identifies the weakness in the process to build software. It involves activities like document review, test case review, walk-throughs, inspection, etc.

Ques.5. What is Quality Control and what are the different types of testing involved in QC?

Ans. Quality control is a product-driven approach that checks that the developed product conforms to all the specified requirements. It is considered a corrective measure as it tests the built product to find the defects. It involves different types of testing like functional testing, performance testing, usability testing, etc.

Ques.6. What is the difference between Verification and Validation?

Ans. The following are the major differences between verification and validation-

#	**Verification**	**Validation**
1.	Verification is the process of evaluating the different artifacts as well as the process of software development. This is done in order to ensure that the product being developed will comply with the standards.	Validation is the process of validating that the developed software product conforms to the specified business requirements.
2.	It is a static process of analyzing the documents and not the actual end product.	It involves dynamic testing of a software product by running it.
3.	Verification is a process-oriented approach.	Validation is a product-oriented approach.

4.	Answers the question – "Are we building the product right?"	Answers the question – "Are we building the right product?"
5.	Errors found during verification require lesser cost/resources to get fixed than those found during the validation phase.	Errors found during validation require more cost/resources. Later the error is discovered higher is the cost to fix it.

Ques.7. What is SDLC?

Ans. SDLC stands for Software Development Life Cycle. It refers to all the activities performed during software development – requirement gathering, requirement analysis, designing, coding or implementation, testing, deployment, and maintenance.

Ques.8. Explain the STLC – Software Testing life cycle.

Ans. The software testing life cycle refers to all the activities performed during the testing of a software product. The phases include-

- **Requirement analysis and validation** – In this phase, the requirements documents are analyzed and validated and the scope of testing is defined.
- **Test planning** – In this phase, the test plan strategy is defined, the estimation of test effort is defined along with the automation strategy, and tool selection is done.
- **Test Design and Analysis** – Here, test cases are designed, test data is prepared and automation scripts are implemented.
- **Test environment setup** – A test environment closely simulating the real-world environment is prepared.
- **Test execution** – The test cases are prepared, bugs are reported and retested once resolved.
- **Test closure and reporting –** A test closure report is prepared to have the final test results summary, learning, and test metrics.

Ques.9. What are the different types of testing?

Ans. Testing can broadly be defined into two types-

- **Functional testing** – Functional testing involves validating the functional specifications of the system.
- **Non Functional testing** – Non-functional testing is a type of testing that involves testing of non-functional requirements of

the system such as performance, scalability, security, endurance, portability, etc.

Going by the way the testing is done, it can be categorized as-

- Black-box testing – In black-box testing, the tester need not have any knowledge of the internal architecture or implementation of the system. The tester interacts with the system through the interface providing input and validating the received output.

- White box testing – In white box testing, the tester analyses the internal architecture of the system as well as the quality of source code on different parameters like code optimization, code coverage, reusability, etc.

- Gray box testing – In gray box testing, the tester has partial access to the internal architecture of the system e.g. the tester may have access to the design documents or database structure. This information helps the tester to test the application better.

Ques.10. What is manual testing?

Ans. Manual testing is a type of testing that involves validation of the requirements of the application by executing a predefined set of test cases manually without the use of any automation tool.

Ques.11. What is automation testing?

Ans. Automation testing is a type of software testing that involves automated test case execution using an automation tool. It helps in reducing the test execution time as the test scripts written once, can be run automatically any number of times without any human intervention.

Ques.12. What are some advantages of automation testing?

Ans. Some advantages of automation testing are-

1. Test execution using automation is fast and saves a considerable amount of time.
2. Carefully written test scripts remove the chance of human error during testing.

3. Test execution can be scheduled for a nightly run using CI tools like Jenkins which can also be configured to provide daily test results to relevant stakeholders.
4. Automation testing is very less resource-intensive. Once the tests are automated, test execution requires almost no time of QAs. Saving QA bandwidth for other exploratory tasks.

Ques.13. What are some disadvantages of automation testing?
Ans. Some disadvantages of automation testing are-

1. It requires skilled automation testing experts to write test scripts.
2. Additional effort to write scripts is required upfront.
3. Automation scripts are limited to verification of the tests that are coded. These tests may miss some error that is very glaring and easily identifiable to human(manual QA).
4. Even with some minor change in the application, script update and maintenance is required.

Ques.14. What is performance testing?
Ans. Performance testing is a type of non-functional testing in which the performance of the system is evaluated under expected or higher load. The various performance parameters evaluated during performance testing are – response time, reliability, resource usage, scalability, etc. The different types of performance testing are – Load, Stress, Endurance, Spike, and Volume Testi
Ques.15. What is a test bed?
Ans. A test bed is a test environment used for testing an application. A test-bed configuration can consist of the hardware and software requirements of the application under test including – operating system, hardware configurations, software configurations, tomcat, database, etc.
Ques.16. What is a test plan?
Ans. A test plan is a formal document describing the scope of testing, the approach to be used, the resources required, and the time estimate for carrying out the testing process. It is derived from the requirement documents (Software Requirement Specifications).

Ques.17. What is a test scenario?
Ans. A test scenario is derived from a use case. It is used for end-to-end testing of a feature of an application. A single test scenario can cater to

multiple test cases. Scenario testing is particularly useful when there is a time constraint while testing.

Ques.18. What is a Test case?
Ans. A test case is used to test the conformance of an application with its requirement specifications. It is a set of conditions with pre-requisites, input values, and expected results in a documented form.

Ques.19. What are some attributes of a test case?
Ans. A test case can have the following attributes-

1. TestCaseId – A unique identifier of the test case.
2. Test Summary – One-liner summary of the test case.
3. Description – Detailed description of the test case.
4. Prerequisite or pre-condition – A set of prerequisites that must be followed before executing the test steps.
5. Test Steps – Detailed steps for performing the test case.
6. Expected result – The expected result in order to pass the test.
7. Actual result – The actual result after executing the test steps.
8. Test Result – Pass/Fail status of the test execution.
9. Automation Status – Identifier of automation – whether the application is automated or not.
10. Date – The test execution date.
11. Executed by – Name of the person executing the test case.

Ques.20. What is Test data?
Ans. Test data is data that is used to test the software with different inputs and helps to check whether the corresponding output is as per the expected result or not. This data is created based on the business requirements.

Ques.21. What is a Test script?
Ans. A test script is an automated test case written in any programming or scripting language. These are basically a set of instructions to evaluate the functioning of an application.

Ques.22. What is an Error in Software Testing?
Ans Since we all are humans so it is obvious to make a mistake. Likewise, an error is a similar case that happens in software testing due to some missing scenario in the requirements, some issues in design, or some mistakes in the implementation.

Ques.23. What is a Bug?
Ans. A bug is a fault in a software product **detected at the time of testing**, causing it to function in an unanticipated manner.

Ques.24. What is a defect?
Ans. A defect is non-conformance with the requirement of the product **detected in production** (after the product goes live).

Ques.25. What are some defect reporting attributes?
Ans. Some of the attributes of a Defect report are-

- DefectId – A unique identifier of the defect.
- Defect Summary – A one-line summary of the defect, more like a defect title.
- Defect Description – A detailed description of the defect.
- Steps to reproduce – The steps to reproduce the defect.
- Expected Result – The expected behavior from which the application is deviating because of the defect.
- Actual Result- The current erroneous state of the application w.r.t. the defect.
- Defect Severity – Based on the criticality of the defect, this field can be set to minor, medium, major, or show stopper.
- Priority – Based on the urgency of the defect, this field can be set on a scale of P0 to P3.

Ques.26. What are some of the bug or defect management tools?
Ans. Some of the most widely used Defect Management tools are – Jira, Bugzilla, Redmine, Mantis, Quality Center, etc.

Ques.27. What is defect density?
Ans. Defect density is the measure of the density of the defects in the system. It can be calculated by dividing the number of defects identified by the total number of lines of code(or methods or classes) in the application or program.

Ques.28. What is defect priority?
Ans. A defect priority is an urgency of fixing the defect. Normally the defect priority is set on a scale of P0 to P3 with the P0 defect having the most urgency to fix.

Ques.29. What is defect severity?
Ans. Defect severity is the severity of the defect impacting the functionality. Based on the organization, we can have different levels of defect severity ranging from minor to critical or show stopper.

Ques.30. Give an example of Low priority-Low severity, Low priority-High severity, High priority-Low severity, and High priority-High severity defects.
Ans. Below are examples for different combinations of priority and severity-

1. **Low priority-Low severity** – A spelling mistake in a page not frequently navigated by users.
2. **Low priority-High severity** – Application crashing in some very corner cases.
3. **High priority-Low severity** – Slight change in logo color or spelling mistake in the company name.
4. **High priority-High severity** – Issue with login functionality.

For details, check – Priority & Severity with Examples

Ques.31. What is a blocker?
Ans. A blocker is a bug of high priority and high severity. It prevents or blocks testing of some other major portion of the application as well.

Ques.32. What is a critical bug?
Ans. A critical bug is a bug that impacts a major functionality of the application and the application cannot be delivered without fixing the bug. It is different from the blocker bug as it doesn't affect or block the testing of other parts of the application.

Ques.33. Explain the bug life cycle or the different states of a bug.
Ans. A bug goes through the following phases in software development-

- New – A bug or defect when detected is in a New state.
- Assigned – The newly detected bug when assigned to the corresponding developer is in the Assigned state.
- Open – When the developer works on the bug, the bug lies in the Open state.
- Rejected/Not a bug – A bug lies in rejected state in case the developer feels the bug is not genuine.

- Deferred – A deferred bug is one whose fix gets deferred for some time(for the next releases) based on the urgency and criticality of the bug.
- Fixed – When a bug is resolved by the developer it is marked as fixed.
- Test – When fixed the bug is assigned to the tester and during this time the bug is marked as in Test.
- Reopened – If the tester is not satisfied with the issue resolution the bug is moved to the Reopened state.
- Verified – After the Test phase, if the tester feels the bug is resolved, it is marked as verified.
- Closed – After the bug is verified, it is moved to Closed status.

Ques.34. What are the different test design techniques?

Ans. Test design techniques are different standards of test designing that allow systematic and widely accepted test cases. The different test design techniques can be categorized as static test design techniques and dynamic test design techniques.

1. Static Test Design Techniques – The test design techniques which involve testing without executing the code. The various static test design techniques can be further divided into two parts manual and using tools-

    - Manual static design techniques
        - Walkthrough
        - Informal reviews
        - Technical reviews
        - Audit
        - Inspection
        - Management review
    - Static design techniques using tools
        - Static analysis of code – It includes analysis of the different paths and flows in the application and different states of the test data.

- Compliance with coding standards – This evaluates the compliance of the code with the different coding standards.
- Analysis of code metrics – The tool used for static analysis is required to evaluate the different metrics like lines of code, complexity, code coverage, etc.

2. Dynamic Test Design Techniques – Dynamic test design techniques involve testing by running the system under test.
    - Specification-based – Specification-based test design techniques are also referred to as black-box testing. These involve testing based on the specification of the system under test without knowing its internal architecture.
    - Structure-based – Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of the code or internal architecture of the system is required to carry out the testing.
    - Experienced-based – The experienced-based techniques are completely based on the experience or intuition of the tester. The two most common forms of experienced-based testing are – Adhoc testing and exploratory testing.

Ques.35. What is Static Testing?

Ans. Static testing is a kind of testing for reviewing the work products or documentation that are being created throughout the entire project. It allows for reviewing the specifications, business requirements, documentation, processes, and functional requirements in the initial phase of testing.

So that the testers involved in it can understand the requirements in more detail before starting the testing lifecycle which intends to help in delivering a quality product.

Ques.36. What is Dynamic Testing?

Ans. The type of testing performed by executing or running the application under test either manually or using automation is called dynamic testing.

Ques.37. Explain the different types of specification-based test design techniques.

Ans. Specification-based test design techniques are also referred to as black-box testing. It involves testing based on the specification of the system under test without knowing its internal architecture. The different types of specification-based test design or black box testing techniques are-

- Equivalence partitioning – Grouping test data into logical groups or equivalence classes with the assumption that all the data items lying in the classes will have the same effect on the application.
- Boundary value analysis – Testing using the boundary values of the equivalence classes taken as the test input.
- Decision tables – Testing using decision tables showing the application's behavior based on a different combination of input values.
- Cause-effect graph – Testing using a graphical representation of the result or outcome and all the factors that affect the outcome.
- State transition testing – Testing based on the state machine model.
- Use case testing – Testing carried out using use cases.

Ques.38. Explain equivalence class partitioning.

Ans. Equivalence class partitioning is a specification-based black-box testing technique. In equivalence class partitioning, a set of input data that defines different test conditions are partitioned into logically similar groups such that using even a single test data from the group for testing can be considered similar to using all the other data in that group.

For example, for testing a Square program (a program that prints the square of a number), the equivalence classes can be-
Set of Negative numbers, whole numbers, decimal numbers, sets of large numbers, etc.

Ques.39. What is boundary value analysis?
Ans. Boundary value analysis is a software testing technique for designing test cases wherein the boundary values of the classes of the equivalence class partitioning are taken as input to the test cases e.g. if the test data lies in the range of 0-100, the boundary value analysis will include test data – 0,1, 99, 100.

Ques.40. What is decision table testing?
Ans. Decision table testing is a type of specification-based test design technique or black-box testing technique in which testing is carried out using decision tables showing the application's behavior based on different combinations of input values.

Decision tables are particularly helpful in designing test cases for complex business scenarios involving the verification of applications with multiple combinations of input.

Ques.41. What is a cause-effect graph?
Ans. A cause-effect graph testing is a black-box test design technique in which a graphical representation of input i.e. cause and output i.e. effect is used for test designing. This technique uses different notations representing AND, OR, NOT, etc relations between the input conditions leading to output.

Ques.42. What is state transition testing?
Ans. State transition testing is a black box test design technique based on a state machine model. State transition testing is based on the concept that a system can be defined as a collection of multiple states and the transition from one state to another happens because of some event.

Ques.43. What is use case testing?
Ans. Use case testing is a black-box testing approach in which testing is carried out using use cases. A use-case scenario is seen as an interaction between the application and actors(users). These use cases are used for depicting requirements and hence can also serve as a basis for acceptance testing.

Ques.44. What is Test Coverage?
Ans. It is a metric that measures the amount of testing performed on software while executing the test cases. Test coverage for any software can

be calculated as the percentage of the number of test areas or coverage items covered with respect to the total number of test areas.

The higher the test coverage, the more the part of the software gets covered by test cases and hence, the more effective will be the testing.

Ques.45. What is structure-based testing?
Ans. Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of the code or internal architecture of the system is required to carry out the testing. The various kinds of testing structure-based or white testing techniques are-

- **Statement testing** – A white box testing technique in which the test scripts are designed to execute the application's code statements. Its coverage is measured as the line of code or statements executed by test scripts.

- **Decision testing/branch testing** – A testing technique in the test scripts is designed to execute the different decision branches (e.g. if-else conditions) in the application. Its coverage is measured as the percentage of decision points out of the total decision points in the application.

- Condition testing – Condition testing is a testing approach in which we test the application with both True and False outcomes for each condition. Hence for n conditions, we will have 2n test scripts.

- **Multiple condition testing** – In multiple condition testing, the different combinations of condition outcomes are tested at least once. Hence for 100% coverage, we will have 2^n test scripts. This is very exhaustive and very difficult to achieve 100% coverage.

- **Condition determination testing** – It is an optimized way of multiple condition testing in which the combinations which don't affect the outcomes are discarded.

- **Path testing** – Testing the independent paths in the system(paths are executable statements from entry to exit points).

Ques.46. What is code coverage?
Ans. Code coverage is the measure of the amount of code covered by the test scripts. It gives the idea of the part of the application covered by the test suite.

Ques.47. What are Statement testing and statement coverage in white box testing?
Ans. Statement testing is a white box testing approach in which test scripts are designed to execute code statements.

Statement coverage is the measure of the percentage of statements of code executed by the test scripts out of the total code statements in the application. The statement coverage is the least preferred metric for checking test coverage.

Ques.48. What is decision testing or branch testing?
Ans. Decision testing or branch testing is a white box testing approach in which test coverage is measured by the percentage of decision points(e.g. if-else conditions) executed out of the total decision points in the application.

Ques.49. What are the different levels of testing?
Ans. Testing can be performed at different levels during the development process. Performing testing activities at multiple levels helps in the early identification of bugs. The different levels of testing are –

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Ques.50. What is unit testing?
Ans. Unit testing is the first level of testing and it involves testing individual modules of the software. It is usually performed by developers.

Ques.51. What is integration testing?
Ans. Integration testing is performed after unit testing. In integration testing, we test the group of related modules. It aims at finding interfacing issues between the modules.

Ques.52. What are the different types of integration testing?
Ans. The different type of integration testing is-

1. Big bang Integration Testing – In big bang integration testing, testing starts only after all the modules are integrated.

2. Top-down Integration Testing – In top-down integration, testing/integration starts from top modules to lower-level modules.

3. Bottom-up Integration Testing – In bottom-up integration, testing starts from lower-level modules to higher-level modules up in the hierarchy.

4. Hybrid Integration Testing – Hybrid integration testing is the combination of both Top-down and bottom-up integration testing. In this approach, the integration starts from the middle layer, and testing is carried out in both the direction

For details check Integration testing.

Ques.53. What is a stub?
Ans. In the case of top-down integration testing, many times lower-level modules are not developed while beginning testing/integration with top-level modules. In those cases, Stubs or dummy modules are used that simulate the working of modules by providing a hard-coded or expected output based on the input values.

Ques.54. What is a driver?
Ans. In the case of bottom-up integration testing, drivers are used to simulating the working of top-level modules in order to test the related modules lower in the hierarchy.

Ques.55. What is system testing?
Ans. System testing is the level of testing where the complete software is tested as a whole. The conformance of the application with its business requirements is checked in system testing.

Ques.56. What is acceptance testing?
Ans. Acceptance testing is testing performed by the potential end-user or customers to check if the software conforms to the business requirements and can be accepted for use.

Ques.57. What is UAT Testing?
Ans UAT testing is the last phase of the testing lifecycle. Its main focus is to validate that software is working in accordance with business requirements. It also ensures that the application is user-friendly and can handle complex scenarios at its best before releasing the product to real-world users.

Ques.58. What is End-To-End Testing?
Ans. End-to-End testing is a type of testing where the entire application undergoes testing, to test that each functionality of the software is working as expected and there is no loophole remaining in it. It ensures that the application is user-friendly and meets the business requirements.

Ques.59. What is alpha testing?
Ans. Alpha testing is a type of acceptance testing that is performed by testers or the internal employees of the organization at the developer site.

Ques.60. What is beta testing?
Ans. Beta testing is the testing done by end-users at the end user's site. It allows users to provide direct input about the software to the development company.

Ques.61. What is Adhoc Testing?
Ans. Adhoc testing is an unstructured way of testing that is performed without any formal documentation or proper planning.

Ques.62. What is monkey testing?
Ans. Monkey testing is a type of testing that is performed randomly without any predefined test cases or test inputs.

Ques.63. How is monkey testing different from Adhoc testing?
Ans. In the case of Adhoc testing although there are no predefined or documented test cases still testers have an understanding of the application. While in the case of monkey testing testers don't have any understanding of the application.

Ques.64. What is exploratory testing?
Ans. Exploratory testing is a type of testing in which new test cases are added and updated while exploring the system or executing test cases. Unlike scripted testing, test design and execution go parallel in exploratory testing.

Ques.65. What is load testing?
Ans. Load testing is a type of performance testing which aims at finding an application's performance under the expected workload. During load testing, we evaluate the response time, throughput, error rate, etc parameters of the application.

Ques.66. What is stress testing?
Ans. Stress testing is a type of performance testing in which an application's behavior is monitored under a higher workload than expected. Stress testing is done to find memory leaks and the robustness of the application.

Ques.67. What is volume testing?
Ans. Volume testing is a type of performance testing in which the performance of the application is evaluated with a large amount of data. It checks the scalability of the application and helps in the identification of a bottleneck with a high volume of data.

Ques.68. What is endurance testing or Soak testing?
Ans. Endurance testing is a type of performance testing which aims at finding issues like memory leaks when an application is subjected to load tests for a long period of time.

Ques.69. What is spike testing?
Ans. Spike testing is a type of performance testing in which the application's performance is measured while suddenly increasing the number of active users during the load test.

Ques.70. What is UI testing?
Ans. UI or user interface testing is a type of testing that aims at finding Graphical User Interface defects in the application and checks that the GUI conforms to the specifications.

Ques.71. What is usability testing?
Ans. Usability testing is the type of testing that aims at determining the ease of using the application. It aims at uncovering the usability defects in the application.

Ques.72. What is Accessibility testing?
Ans. Accessibility testing is the type of testing which aims at determining

the ease of use or operation of the application specifically for people with disabilities.

Ques.73. What is compatibility testing?
Ans. Compatibility testing is validating software to see how compatible the software is with a particular environment – operating system, platform, or hardware.

Ques.74. What is configuration testing?
Ans. Configuration testing is the type of testing used to evaluate the configurational requirements of the software along with the effect of changing the required configuration.

Ques.75. What is localization testing?
Ans. Localization testing is a type of testing in which we evaluate the application's customization(a localized version of the application) in a particular culture, locale or country.

Ques.76. What is globalization testing?
Ans. Globalization testing is a type of testing in which an application is evaluated for its functioning across the world in different cultures, languages, locales, and countries.

Ques.77. What is negative testing?
Ans. Negative testing is a type of testing in which the application's robustness(graceful exiting or error reporting) is evaluated when provided with invalid input or test data.

Ques.78. What is security testing?
Ans. Security testing is a type of testing which aims at evaluating the integrity, authentication, authorization, availability, confidentiality, and non-repudiation of the application under test.

Ques.79. What is penetration testing?
Ans. Penetration testing or pen testing is a type of security testing in which an application is evaluated(safely exploited) for different kinds of vulnerabilities that any hacker could exploit.

Ques.80. What is robustness testing?
Ans. Robustness testing is a type of testing that is performed to find the

robustness of the application i.e. the ability of the system to behave gracefully in case of erroneous test steps and test input.

Ques.81. What is concurrency testing?
Ans. Concurrency testing is multi-user testing in which an application is evaluated by analyzing the application's behavior with concurrent users accessing the same functionality.

Ques.82. What is backend testing?
Ans. Backend testing is a type of testing that involves testing the backend of the system which comprises testing the databases and the APIs in the application.

Ques.83. What is A/B testing?
Ans. A/B testing is a type of testing in which the two variants of the software product are exposed to the end-users and on analyzing the user behavior on each variant, the better variant is chosen and used thereafter.

Ques.84. What is risk analysis?
Ans. Risk analysis is the analysis of the risk identified and assigning an appropriate risk level to the defect based on its impact on the application.

Ques.85. What is the difference between regression and retesting?
Ans. Regression testing involves testing the application to verify that a new code change doesn't affect the other parts of the application. Whereas, in retesting, we verify if the fixed issue is resolved or not.

Ques.86. What is the difference between black-box and white-box testing?
Ans. Black-box testing is a type of testing in which the internal architecture of the code is not required for testing. It is usually applicable for system and acceptance testing.

Whereas white-box testing requires internal design and implementation knowledge of the application being tested. It is usually applicable for Unit and Integration testing.

Ques.87. What is the difference between smoke and sanity testing?
Ans. The difference between smoke and sanity testing is-

- Smoke testing is a type of testing in which all major functionalities of the application are tested before carrying out exhaustive testing. Whereas, sanity testing is a subset of regression testing which is carried out when there is some minor fix in the application in a new build.
- In smoke testing, shallow-wide testing is carried out while in Sanity, narrow-deep testing (for a particular functionality) is done.
- The smoke tests are usually documented or automated. Whereas, the sanity tests are generally not documented or unscripted.

Ques.88. What is the difference between Release and Build?
Ans. A build is an executable file provided by the developers to the testing team for testing the application. It undergoes various iterations of fixing and testing until the application works as expected. Once the application becomes stable and ready for the end-users, it's released in the market.

Whereas, a release is installable software provided to the end-users after it gets certified by the testing team. During the release of any software to the client, release notes are attached to it that includes a number of defects still open, covered user stories, change requirements, and version of the release.

Manual Testing Interview Questions for Experienced

Ques.89. What is the difference between bug leakage and bug release?
Ans. Bug leakage is when the tested software is released into the market and the end-user finds bugs in it. These include the bugs that got missed by the testing team during the testing phase.

Whereas, bug release is when a specific version of the software is released in the market with some known bugs which are intended to get fixed in

later versions. These types of issues are of low priority and are mentioned in the release notes while sharing with the end-users.

Ques.90. What do you mean by Defect Triage?
Ans. Defect triage is a process in which the defects are prioritized based on different factors like severity, risk, the time required to fix the bug, etc. The defect triage meeting includes the different stakeholders – the development team, testing team, project manager, BAs, etc, which decide the priority of fixing the defects.

Ques.91. What is a test harness? Why do we need a test harness?
Ans. A test harness is a collection of test scripts and test data usually associated with the unit and integration testing. It involves stubs and drivers that are required for testing software modules and integrated components.

Ques.92. What is all pair testing?
Ans. All pair testing is a type of testing in which the application is tested with all possible combinations of the values of input parameters.

Ques.93. What is failover testing?
Ans. Failover testing is a type of testing that is used to verify the application's ability to allocate more resources(more servers) in case of failure and transfer the processing part to the backup system.

Ques.94. What is fuzz testing?
Ans. Fuzz testing is a type of testing in which a large amount of random data is provided as input to the application in order to find security loopholes and other issues in the application.

Ques.95. What is pilot testing?
Ans. Pilot testing is testing carried out as a trial by a limited number of users to evaluate the system and provide their feedback before the complete deployment is carried out.

Ques.96. What is dev-box Testing?
Ans. In dev-box testing, a tester performs testing on the developer's system to verify if the major functionalities of the application are stable and ready for testing.

Ques.97. What is mutation testing?
Ans. Mutation testing is a type of white box testing in which the source code of the application is mutated to cause some defects in its working. After that, the test scripts are executed to check for their correctness by verifying the failures caused by the mutant code.

Ques.98. What is the requirement traceability matrix(RTM)?
Ans. In software testing, a requirement traceability matrix is a table that relates the high-level requirements with detailed requirements, test plans, or test cases. RTM helps in ensuring 100% test coverage.

Ques.99. What is cyclomatic complexity?
Ans. Cyclomatic complexity is the measure of the number of independent paths in an application or program. This metric provides an indication of the amount of effort required to test complete functionality. It can be defined by the expression –
L – N + 2P, where:
L is the number of edges in the graph
N is the number of nodes
P is the number of disconnected parts

Ques.100. What are the entry criteria in software testing?
Ans. A set of prerequisites that are required to kick off the testing activity includes a Test environment, Test tool, Test Data, database connectivity, and many more.

Ques.101. What is the exit criteria in software testing?
Ans. An exit criteria is a formal set of conditions that specify the agreed-upon features or state of the application in order to mark the completion of the process or product.

Ques.102. What is the difference between testing and debugging?
Ans. Testing is primarily performed by the testing team in order to find the defects in the system. Whereas, debugging is an activity performed by the development team. In debugging the cause of the defect is located and fixed. Thus removing the defect and preventing any future occurrence of the defect as well.

Another difference between the two is – testing can be done without any

internal knowledge of software architecture. Whereas debugging requires knowledge of software architecture and coding.

Ques.103. Explain the Agile methodology.
Ans. The agile methodology of software development is based on an iterative and incremental approach. In this model, the application is broken down into smaller builds on which different cross-functional teamwork together, providing rapid delivery along with adapting to changing needs at the same time.

Ques.104. What is scrum?
Ans. A scrum is a process for implementing Agile methodology. In scrum, time is divided into sprints and on completion of sprints, a deliverable is shipped.

Ques.105. What are the different roles in scrum?
Ans. The different roles in scrum are –

1. Product Owner – The product owner owns the whole development of the product, assigns tasks to the team, and acts as an interface between the scrum team(development team) and the stakeholders.
2. Scrum Master – The scrum master monitors that scrum rules get followed in the team and conducts scrum meetings.
3. Scrum Team – A scrum team participates in the scrum meetings and perform the tasks assigned.

Ques.106. What is a scrum meeting?
Ans. A scrum meeting is a daily meeting in the scrum process. This meeting is conducted by the scrum master and an update of the previous day's work along with the next day's task and context is defined in this meeting.

Ques.107. Explain TDD (Test Driven Development).
Ans. Test-Driven Development is a software development methodology in which the development of the software is driven by test cases created for the functionality to be implemented. In TDD, first, the test cases are created, and then the code to pass the tests is written. Later the code is refactored as per the standards.

Ques.108. What is the difference between Latent and Masked Defects?
Ans. A latent defect is an unidentified defect present in the current release

but is not visible because the conditions in which the defect could be found have never been met. These types of defects occur only when a particular event gets triggered which was concealing their presence.

Whereas a masked defect is an existing defect that has not yet caused any failure because another error has masked it or prevented it from getting discovered.

Ques.109. What is the PDCA cycle in software testing?
Ans. PDCA cycle is a key for continuous process improvement in software development. It includes the following 4 steps-

- Plan – Plan the objectives, goals, and initiatives which help to reach customer satisfaction.
- Do – It implements the plan into action. To serve the customer with better quality and satisfaction it is necessary to have a good plan to execute.
- Check – To check the progress of your plan which has been implemented. The result will show how accurate the planning had been done.
- Act – Acting upon the results to do further improvement which helps in achieving the planned goals.

Ques.110. What is Defect Cascading?
And. Defect cascading is the triggering of a defect by another defect. It happens when a defect is not caught by the testing team and it gives rise to another defect.

Ques.111. What is a test metric?
Ans. Test Metric is a quantitative analysis that helps in monitoring the progress of a software project. Every project has its own timeline so ensuring the delivery of the project on time requires setting deliverables at different intervals and this aspect of measuring the progress is provided by test metrics.

Ques.112. What is Context-driven testing?
Ans. Context-driven testing is the type of testing that involves adopting the test practices, and methodologies and at times customizing them based on

the context of the project.

Ques.113. How can you test an application without a requirement document?

Ans. An application can be tested without a formal requirement document by using the following measures-

- By researching applications that are similar in nature.
- Testing based on ideal user experience. Even without having any requirements, a tester can still test the application's user-friendliness.
- Using exploratory testing. Since there is no documentation involved, so testers can use exploratory testing and test the application on the fly with a more hands-on approach.
- Asking as many questions as possible and brainstorming with the different stakeholders – product managers, developers, etc.

Ques.114. How to write efficient test cases?

Ans. We can write efficient test cases by using below approaches-

- Following test design techniques like boundary value analysis, equivalence class partitioning, decision table testing, etc.
- By writing clear and concise test cases that are not ambiguous.
- Following a uniform nomenclature also helps in creating efficient test cases.
- Avoiding redundancy leads to the wastage of both resources and time. So, good test cases need not have any redundancy.
- Using a requirement traceability matrix helps in ensuring that the test coverage is maximized.