Depending on the objective of testing

1. Functional testing – Functional testing is a type of testing that involves validating the application with its functional specifications or business requirements.

2. Non-functional testing – Non-functional testing is a type of testing that includes testing the non-functional attributes or requirements of the system like performance, reliability, security, scalability, usability etc.

Based on test execution method – manual or automated

3. Manual testing

Manual testing is a type of testing in which test case execution is performed manually by humans.

4. Automated testing – Automation testing is a type of testing in which automated test case execution is performed using different automation tools and scripts.

Based on test design technique

5. Black box testing – Black box testing is a type of testing in which the tester is not required to have any knowledge of the internal architecture or implementation of the system.

6. White box testing – In the case of white box testing, the tester needs to have access and knowledge of the internal architecture of the application. The tester analyses the architecture as well as the source code on different quality parameters like code coverage, code optimization, reusability etc.

7. Glass box testing – Same as white box testing

8. Gray box testing – In Gray box testing, the tester has limited access/knowledge of the internal architecture of the system e.g. the tester might not have access to the whole source code of the application but may have access to the design documents or the structure of the database(schema and tables). All this information helps the tester in testing the application better.

9. Specification-based testing – Same as black-box testing.

10. Structure-based testing – Same as white-box testing.

Levels of testing

11. Unit testing – Unit Testing is the first level of testing usually performed by the developers. In unit testing, a module or component is tested in isolation.

12. Integration testing – Integration testing is the testing of a group of related modules. It aims at finding interfacing issues between the modules.

13. System testing – System testing is the level of testing where the complete integrated application is tested as a whole. It aims at determining if the application conforms to its business requirements.

14. Acceptance testing – Acceptance testing is the final and one of the most important levels of testing on successful completion of which the application is released to production.

Type of integration testing

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

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

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

18. 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 middle layer and testing is carried out in both the direction

Type of acceptance testing

19. Alpha testing – Alpha testing is a type of acceptance testing that is performed end-users at the developer site.

20. Beta testing – Beta testing is the testing done by end-users at end user’s site. It allows users to provide direct input about the software to the development company.

Black box testing or specification based testing types

21. Equivalence partitioning testing – Grouping test data into logical groups or equivalence classes with the assumption that any all the data items lying in the classes will have the same effect on the application.

22. Boundary value analysis testing – Testing using the boundary values of the equivalence classes taken as the test input.

23. Decision tables testing – Testing using decision tables showing application’s behavior based on different combination of input values.

24. Cause-effect graph testing – Testing using a graphical representation of input i.e. cause and output i.e. effect is used for test designing.

25. State transition testing – Testing based on the state machine model.

26. Use case testing – Testing carried out using use cases.

White-box or structural testing types

{These testing types are performed by devs and white-box testers only, feel free to ignore these in case you are not into white-box testing}

27. Statement testing – Test scripts are designed to execute code statements and coverage is the measure of line of code or statements executed by test scripts.

28. Decision testing/branch testing – Measure of the percentage of decision points(e.g. if-else conditions) executed out of the total decision points in the application.

29. Condition testing – Testing the condition outcomes(TRUE or FALSE). So, getting 100% condition coverage required exercising each condition for both TRUE and FALSE results using test scripts(For n conditions we will have 2n test scripts).

30. Multiple condition testing – Testing the different combinations of condition outcomes. Hence for 100% coverage we will have 2^n test scripts. This is very exhaustive and very difficult to achieve 100% coverage.

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

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

33. Mutation testing – Mutation testing is a type of white box testing in which the application’s source code is intentionally mutated (changed) in order to cause a defect. After that test scripts are executed and verified if they fail due to the change in code.

34. Loop testing – Loop testing is a type of white box testing in which primarily focuses on validating the different kinds of loop constructs.

Performance testing types

35. Performance testing – Performance testing is type of non-functional testing that is performed to evaluate the different performance attributes of the system like – responsiveness, stability, reliability etc in a particular workload.

36. Load testing – Load testing is a type of performance testing which involves evaluating the performance of the system under expected workload. A typical load test includes determining the different performance parameters like response time, throughput, error rate etc during the course of the load test.

37. Stress testing – Stress testing is a type of performance testing in which we evaluate application’s performance at load much higher than the expected load. In stress testing we determine the breakpoint of the application, the point at which the application fails to respond in correct manner.

38. Endurance testing – Endurance testing or ‘Soak Testing’ is a type of testing performed to evaluate if the system can sustain continuous expected load for very long durations (spanning days). It helps in finding issues like memory leakage.

39. Soak testing – Same as endurance testing.

40. Stability testing – Same as endurance testing

41. Spike testing – In spike testing, the behavior of the application is observed with sudden increase in the number of users. It helps in checking if the application can recover on sudden increment/decrement in the number of active users.

42. Volume testing – Volume testing is a type of testing in which the application is subjected to a very high volume of data. This is performed either by inserting large amount of data in the database or bypassing large file to the application for processing.

Experience based testing

43. Experience-based testing – The experienced-based testing techniques are completely based on the experience or intuition of the tester. Two most common forms of experienced-based testing are – Adhoc testing and Exploratory testing.

44. Adhoc testing – Adhoc testing is an unstructured way of testing that is performed without any formal documentation or proper planning.

45. Exploratory testing – Exploratory testing is a type of testing in which new test case are added and updated while exploring the system or executing test cases. Unlike scripted testing, test design and execution goes in parallel during Exploratory testing.

All other types of testing

46. Retesting – Retesting is a type of testing in which we verify if the fixed issue is resolved or not.

47. Regression testing – Regression testing includes testing the application to verify that a new code change doesn’t affect the other parts of the application.

48. Smoke testing – Smoke testing is a type of testing in which the all major functionalities of the application are tested before carrying out exhaustive testing.

49. Sanity testing – Sanity testing is the subset of regression testing, which is carried out when there is some minor fix in the application in a new build.

50. Dynamic testing – Testing performed by executing or running the application under test either manually or using automation (opposite of reviews and walkthrough).

51. Static testing – Static testing is a type of testing in which testing is carried out without actually running the code. It involves activities like reviews, inspections, and walk-through.

52. Monkey testing – Monkey testing is a type of testing that is performed randomly without any predefined test cases or test inputs.

53. Gorilla testing – Gorilla testing involves testing an individual module or functionality of the application heavily in order to test its robustness.

Usability testing – Usability testing is the type of testing that aims at determining the extent to which the application is easy to understand and use.

54. Accessibility testing – Accessibility is the type of testing which aims at determining the ease of use or operation of the application specifically for the people with disabilities.

55. Compatibility testing – Testing software to see how compatible the software is with a particular environment – Operating system, platform or hardware.

56. Configuration testing – Configuration testing is the type of testing used to evaluate the configuration requirements of the software along with effect of changing the required configuration.

57. Localization testing – Localisation testing is a type of testing in which we evaluate the application’s customization(a localized version of the application) to a particular culture or locale. Generally the content of the application is checked for updation(e.g. content language).

58. Globalization testing – Globalisation testing is a type of testing in which application is evaluated for its functioning across the world.

59. Internationalization testing – Same as globalisation testing

60. Negative testing – 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.

61. Security testing – 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.

62. Penetration testing – Penetration testing or pen testing is a type of security testing in which application is evaluated(safely exploited) for different kinds of vulnerabilities that any hacker could exploit.

63. Crowdsourced testing – Crowd-sourced testing is a type of testing which is carried out by a large group or community of QA professionals instead of in-house QAs or hired QA consultants.

64. Database testing – Database testing is a type of testing that involves checking the integrity of actual data in the front end with the data present in the database. It involves validating the data in the database, checking that there are no orphan records (record with a foreign key to a parent record that has been deleted”), no junk records are present, updating records in the database and verify the value in the front end.

65. API testing – API testing is a type of testing that involves testing of the Restful APIs and SOAP web services directly using some client like Advanced Rest Client or tools like SOAPUI.

66. ETL testing – ETL(Extract-Transform-Load) testing is a type of testing that involves checking the consistency of data after extraction from source to destination.

67. Data warehouse testing – Same as ETL testing.

68. Robustness testing – 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.

69. A/B testing – 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.

70. Concurrency testing – Concurrency testing is a multi-user testing in which an application is evaluated by analyzing application’s behavior with concurrent users accessing the same functionality.

71. All pair testing – All pair testing is a type of testing in which the application is tested with all possible combination of the values of input parameters.

72. Failover testing – Failover testing is a type of testing that is used to verify application’s ability to allocate more resources(more servers) in case of failure and transferring the processing part to back-up system.

73. Fuzz testing – Fuzz testing is a type of testing in which large amount of random data is provided as input to the application in order to find security loopholes and other issues in the application.

73. Fault injection testing – Fault injection testing is a type of testing in which fault is intentionally introduced in the application in order to improve the test coverage.

74. UI testing – 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.

75. Pilot testing – Pilot testing is a testing carried out as a trial by limited number of users evaluate the system and provide their feedback before the complete deployment is carried out.

76. Backend testing – Backend testing is a type of testing that involves testing the back-end of the system which comprises of testing the databases and the APIs in the application.

77. Compatibility testing – Compatibility testing is a type of non-functional testing that involves checking the compatibility of the application with different environmental factors like – operating system, hardware, browser, network, devices, software versions, etc.

78. Browser compatibility testing – Browser compatibility testing is a type of testing that involves validating the correctness and consistency of the application over different browsers.

79. Cross-browser testing – Same as browser compatibility testing.

80. Forward compatibility Testing – Forward compatibility testing involves validating the application with newer version of the other platform or software.

81. Backward compatibility testing – Backward compatibility testing involves validating the application with an older version of the other platform or software.

82. Downward compatibility testing – Same as Backward compatibility testing

83. Component testing – Component testing is a type of testing that involves testing the functionality of the individual components or module of the application.

84. Module testing – Same as component testing

85. Agile Testing – Agile testing is a type of testing that involves following principles of agile software development methodology. In this agile testing, testing is conducted throughout the life cycle of the continuously evolving project instead of being confined to a particular phase.

86. End to end testing – End to end testing is a type of testing in which the application’s flow is tested from start to end under real-world scenarios in order to ensure that application works as per the requirement.

87. Happy path testing – Testing the positive or default flow of the application with valid input is called happy path testing.

88. Incremental testing – Incremental testing is also known as incremental integration testing. In this testing, integration between the modules is tested and on successful testing new modules are incrementally added till the time each module of the application is integrated and tested.

89. Recovery testing – Recovery testing is a type of non-functional testing that involves testing the ability of the system to recover from a crash or failure.

90. Risk-based testing – Risk-based testing is a type of testing in which test cases are prioritized on the basis of risk involved or the impact of the failure.

91. Vulnerability testing – Vulnerability testing involves testing and identification of the vulnerability or weakness in the application.

92. Compliance testing – Compliance testing is a type of non-functional testing that involves validating that the built product conforms the organization’s standards and other

93. Conformance testing – Same as compliance testing

94. Destructive testing – A type of testing that aims at checking the robustness of the application by intentionally breaking or crashing the application.

95. Dependency testing – Dependency testing involves checking the pre-conditions or the initial state and configuration of the system required for correct functioning of the application.

96. Scenario testing – Scenario testing is a type of testing in which testing of complex test flows is simplified by making use of scenarios i.e. user flows. It helps in testing the end to end flow of the application.

97. Installation testing – In installation testing, the installation process is checked based on the installation guide.

98. Documentation testing – Documentation testing involves reviewing and validating the documents and user manual of the system.

99. AB testing – AB testing is a way of creating and analyzing two variants of an application in order to find which variant performs better in terms of user experience, monetary benefits(e.g. for e-commerce application) or any xyz goal. And then eventually keeping the better performing variant.

100. Split testing – Same as AB testing

With this, we have come to the end of this article. If you have any questions, please ask in the comment section.Also,check out complete software testing tutorial below.