

170 plus Software Testing Types

The following are the different types of Software Testing:

1) Manual Testing

Manual Testing is a type of Software Testing where humans perform the testing or check the working of the Software by sitting in-front of their computer machines or mobile devices without using any Test Automation Tools.

2) Automation Testing

In Automation Testing, the Test Automation Tools will test the Software instead of humans. Test Automation tool uses the automation code written by the human or tester to perform the testing on behalf of the human.

3) Functional Testing

Functional Testing is performed to test whether the software application is working or functioning according to the specified business requirement.

4) Non-Functional Testing

Non-Functional Testing is performed to test or check the non-functional characteristics of the Software like Performance, Security etc.

5) Black Box Testing

Black Box Testing is performed to test the visual functioning of the software without looking into its internal code structure, design and implementation.

6) White Box Testing

White Box Testing is a type of Software Testing, which tests the internal code and program structure of the Software.

7) Positive Testing

Positive Testing is a process of testing which is performed to verify the working of the application by providing valid inputs.

8) Negative Testing

Negative Testing is a process of testing which is performed to verify whether the application is able to handle invalid inputs properly.

9) Performance Testing

Performance Testing is a Non-Functional Testing, which is performed to test the speed and stability of the application under different work loads.

10) Security Testing

Security Testing is a Non-Functional Testing, which is performed to identify/uncover the security loopholes (i.e. Vulnerabilities / Security Bugs) in the given software system.

11) Smoke Testing

Smoke Testing is a type of testing in which only the major or critical functionalities of the software are tested on the provided build before moving to the detailed testing phase.

12) Regression Testing

Regression testing is a type of testing performed on the previous working functionality, to find out any issues that may have got introduced by newly adding or updating the application code base.

13) Retesting

Retesting is a type of testing which is performed to verify whether the defects reported earlier are fixed by the developer.

14) Sanity Testing

Sanity Testing is performed to verify whether the minor bug fixes or minor changes (which are done on the stable software) are working fine, by only verifying the changed functionality and its related functionality in depth.

15) Unit Testing

Unit Testing is one of the levels of Software Testing, where the testing is performed on the isolated smallest units of Software Application programs. Unit in Unit Testing is nothing but the smallest piece of software code which is testable.

16) Integration Testing

Integration Testing is one of the Software Testing Levels, in which the individual tested units are integrated (i.e. combined) and tested in groups.

17) System Testing

System Testing is one of the Software Testing levels, in which the complete integrated system is tested to verify the behavior against specified requirements.

18) Acceptance Testing

Acceptance Testing is the final level of Software Testing. Once the testing team assesses the product by performing System Testing, it needs to be handed over to Customer or end users for its acceptability. Hence Acceptance Testing will be performed by the Customers or end users in-order to accept or reject the product/application delivered by the testing team. As part of Acceptance Testing, the delivered application/product will be verified against the business requirements along with end-to-end real-time business flows in a separate Test Environment mimicking production like data.

19) Ad-hoc Testing

Ad-hoc Testing is an informal and unplanned type of testing, where testers randomly perform testing on any part of the application without using any formal test plans/test cases/Other documents.

20) Agile Testing

Agile Testing in simple terms is a testing practice which follows the principles involved in Agile Software Development model while testing the software.

21) Usability Testing

Usability Testing checks whether the application is easy to understand by having user friendly GUI and easy to navigate for smooth interaction of users with the application.

22) User Acceptance Testing

User Acceptance Testing or Acceptance Testing is the final level of Software Testing. Once the testing team assesses the product by performing System Testing, it needs to be handed over to Customer or end users for its acceptability. Hence Acceptance Testing will be performed by the Customers or end users in-order to accept or reject the product/application delivered by the testing team. As part of Acceptance Testing, the delivered application/product will be verified against the business requirements along with end-to-end real-time business flows in a separate Test Environment mimicking production like data.

23) UI Testing (User Interface Testing)

UI Testing is a type of testing where we test whether the Graphical User Interface of the application is according to the requirement specifications.

24) Static Testing

Static Testing is a software testing technique in which the testing is performed without executing the developed code. In this testing technique, the testing team will manually examine the developed code, requirement documents, and design documents to find the defects.

25) Dynamic Testing

Dynamic Testing is a software testing technique in which the testing is performed by executing the developed code (i.e. The Running Application) to validate the dynamic behavior of the code.

26) Pair Testing

Pair Testing is a type of testing where, two people test the same functional of an application at the same time. This allows them to generate and share more ideas while testing resulting in better test coverage.

27) Mutation Testing

Mutation Testing is a white box testing technique in which the application source code is mutated (changed) in small ways, to check the quality of the test cases by verifying that the faulty mutated code is failing the test cases.

28) Mobile Testing

Mobile Testing is nothing but testing the application that is developed for working on any mobile devices. Testing of the application will be performed on the supported mobile devices.

29) Monkey Testing

Money Testing is a type of testing, where the application is randomly tested with random input data.

30) Fuzz Testing

Fuzz Testing is a type of Security Testing, where the tester tries to identify the security loopholes in the application by breaking into the application using fuzz data (i.e. invalid random data).

31) Compatibility Testing

Compatibility Testing is performed to test whether the given application is working correctly on the supported browsers, operating systems, hardware, mobile devices, databases and other compatible software.

32) Database Testing

Database Testing is a type of testing, where we check the data that is displayed on the Graphical User Interface of the application is matching with the data records that are stored in the application's database.

33) End-to-End Testing

Every complex Software system comprises of several sub-systems. End to End Testing is performed to check the working of the real-time end-to-end user scenarios/flows covering multiple interconnected sub-systems inside the whole system.

34) ETL Testing

ETL (Extract Transform Load) is used in the Data Warehousing projects for performing the below:

>> Extracting the data from different data sources

>> Transforming or applying calculations on the extracted data as per the destination requirement

>> Loads the transformed data into a Data Warehouse.

35) Extreme Testing

Extreme Testing is the approach for testing the applications which are getting developed using Extreme Programming (XP) development methodology.

36) Load Testing

Load Testing is a type of performance testing, which is performed to check whether the application can handle the real world load conditions (i.e. normal user load and peak user load).

37) Stress Testing

Stress Testing is a type of performance testing, where the load beyond the applications capability is applied to check the breaking point of the application and its recovering capability after breaking.

38) Endurance Testing

Endurance Testing is a type of performance testing, where the application is tested with an expected load for a long period of time to check whether the application can handle the load for that period.

39) Soak Testing

Soak Testing is also known as Endurance Testing. Hence Soak Testing is a type of performance testing, where the application is tested with an expected load for a long period of time to check whether the application can handle the load for that period.

40) Capacity Testing

Capacity Testing is also known as Endurance Testing. Hence Capacity Testing is a type of performance testing, where the application is tested with an expected load for a long period of time to check whether the application can handle the load for that period.

41) Volume Testing

Volume Testing is a type of performance testing, where a large amount of data is added to the database of the application and test whether the application can handle such data.

42) Flood Testing

Flood Testing is also known as Volume Testing. Hence Flood Testing is a type of performance testing, where a large amount of data is added to the database of the application and test whether the application can handle such data.

43) Scalability Testing

Scalability Testing is a type of performance testing, where we test the flexibility of the application to scale up in proportion to the increasing demands of the user work load.

44) Penetration Testing

Penetration Testing is performed to check whether the identified vulnerabilities (i.e. security loopholes or bugs) in the application can be used to exploit/attack/compromise the application.

45) Vulnerability Assessment Testing

Vulnerability Assessment Testing is performed to discover the vulnerabilities (i.e. security loopholes or bugs) in the application, with an intention of protecting the application from future hacks.

46) Grey Box Testing

Grey Box Testing is the combination of Black Box Testing and White Box Testing, where the tester having the partial white box knowledge will design the better test cases for testing the visual functionality of the application. Hence in Grey Box Testing, partial white box testing knowledge will be used for designing the better test cases and black box testing knowledge is used for verifying the visual functionality of the application.

47) Exploratory Testing

Exploratory Testing is a formal/systematic way of testing the application, where the testing is performed by exploring and simultaneously learning the application without using any test plans, test cases or requirement specifications.

48) Context Driven Testing

In Context Driven Testing approach, testing will be performed on the application by choosing the right test techniques, test deliverable, test documentation and test objectives based on the project's context, as what works for one project may not work for other project.

49) Web Services Testing

Web Service in simple terms is the service offered by a web application residing on a machine. Testing the working of these Web Services is known as Web Services Testing.

50) API Testing

API (Application Programming Interface) connect different systems by enabling the data communication between them via the application's programming interface. Testing the Application's programming interfaces is known as API Testing.

51) Alpha Testing

Alpha Testing is a type of internal Acceptance Testing performed on the Application at the development site before the Beta Testing is performed. Alpha Testing will be performed by developers followed by the highly skilled testers with an intention of catching the bugs quickly and their by avoiding the release of buggy application for beta testing where real public will be involved in testing.

52) Beta Testing

Beta Testing is a type of open Acceptance Testing performed on the Application by the real end-users of the application in their real environments.

53) Reliability Testing

Reliability Testing is a type of testing, performed on the application for a specific period of time to check whether the application is fault proof and reliable.

54) Backward Compatibility Testing

Testing which checks whether the newer version of developed application is still able to work with the older version of other software, hardware etc. is known as Backward Compatibility Testing.

55) Forward Compatibility Testing

Testing which checks whether the developed application still works with the newer versions of other software, hardware etc. is known as Forward Comparability Testing.

56) Gorilla Testing

Gorilla Testing is a type of testing, where the functionality of a specific module is tested in an exhaustive manner until the application is proven as robust and error free.

57) Frustration Testing

Frustration Testing is also known as Gorilla Testing, as it is performed in a repetitive and exhaustive manner by testing the functionality of a specific module to its core with an intention of finding the bugs at any cost.

58) Torture Testing

Torture Testing is also known as Gorilla Testing, as it is performed in a repetitive and exhaustive manner by testing the functionality of a specific module to its core with an intention of finding the bugs at any cost.

59) Fault Tolerance Testing

Fault Tolerance Testing is also known as Gorilla Testing, as it is performed in a repetitive and exhaustive manner by testing the functionality of a specific module to its core with an intention of finding the bugs at any cost.

60) Glass Box Testing

Glass Box Testing is also known as White Box Testing, which tests the internal code and program structure of the software.

61) Clear Box Testing

Clear Box Testing is also known as White Box Testing, which tests the internal code and program structure of the software.

62) Open Box Testing

Open Box Testing is also known as White Box Testing, which tests the internal code and program structure of the software.

63) Transparent Box Testing

Transparent Box Testing is also known as White Box Testing, which tests the internal code and program structure of the software.

64) GUI Testing

GUI Testing is a type of testing, where we test the application's Graphical User Interface. GUI Testing is a subset of UI testing, where UI Testing not only tests the GUI of the application but also the CLIs (Command Line Interfaces).

65) Globalization Testing

Globalization Testing is a type of testing, where we check whether the given application is working globally (i.e. independent of its geographical and cultural environment).

66) Internationalization Testing

Internationalization Testing is a type of testing, where the testers check whether the application can adapt to other languages and regions without any changes.\

67) Localization Testing

Localization Testing is performed to test whether the application works for a particular language, region and locale.

68) Installation Testing

Installation Testing is performed to check whether the application is correctly installed (i.e. all required components are correctly installed) and the application is working as expected after installation.

69) Uninstallation Testing

Uninstallation Testing is performed to check whether the application is completely uninstalled (i.e. to check all the components of application are removed during uninstallation along with the files and folder structure)

70) Path Testing

Path Testing is a structural testing method, which uses the source code of the application in order to find all the executable paths of testing and then generate test cases based on the paths found.

71) Basis Path Testing

Basis Path Testing is similar to Path Testing, which derives the test cases by going through the flows and logical paths in the source code of the application.

72) Benchmark Testing

Benchmark Testing is done by the Performance Testing team for checking whether the performance testing metrics of the application is passing the organization wide agreed standard.

73) Baseline Testing

Baseline Testing is done by the Performance Testing team when the application undergoes any changes such as upgrade etc., to check the comparison of the new performance testing metrics against the older baseline metrics which were collected before.

74) Portability Testing

Portability Testing is a type testing, which is performed to find-out the ease with which the application can be moved from one environment to another environment.

75) Binary Portability Testing

Binary Portability Testing is performed to test the portability of an executable application across different supported platforms and environments, usually for the conformation to ABI (Application Binary Interface) specification.

76) Branch Coverage Testing

In Branch Coverage Testing, the testers will make sure that all the possible program branches resulting from each decision point are tested at-least once.

77) Breadth Testing

In Breadth Testing, the testers will test all the features of the application are tested but not in depth.

78) Depth Testing

In Depth Testing, a feature of an application is tested to full extent.

79) Behavioral Testing

Behavioral Testing is also known as Black Box Testing, which is performed to test the visual functioning of the software without looking into its internal code structure, design and implementation.

80) Build Acceptance Testing

Build Acceptance Testing is a basic testing of a new build to check whether the build is testable and stable enough before it is released to the Testing team for detailed testing.

81) Buddy Testing

Buddy Testing is similar to Pair Testing, but in Buddy Testing the combination of Unit and System Testing will be performed together by a developer and a tester.

82) Comparison Testing

Comparison Testing is all about evaluating the strengths and weakness of the application against the applications that already exist as its competition in the market.

83) Code Driven Testing

Code Driven Testing is performed by creating the test for the next development item before developing it. Once the development item for the created test is developed, it will be tested, fixed if failed and retested until the created test passes. Once the test is passed, then the same process is followed by creating another test for the next item to be developed.

84) Component Testing

Component Testing is also known as Unit Testing, which is one of the levels of Software Testing, where the testing is performed on the isolated smallest units of Software Application programs. Unit in Unit Testing is nothing but the smallest piece of software code which is testable.

85) Module Testing

Module Testing is also known as Unit Testing, which is one of the levels of Software Testing, where the testing is performed on the isolated smallest units of Software Application programs. Unit in Unit Testing is nothing but the smallest piece of software code which is testable.

86) Program Testing

Program Testing is also known as Unit Testing, which is one of the levels of Software Testing, where the testing is performed on the isolated smallest units of Software Application programs. Unit in Unit Testing is nothing but the smallest piece of software code which is testable.

87) Configuration Testing

Configuration Testing is performed by testing on different supported system configurations to figure-out the optimal configuration of the application which doesn't result in any bugs.

88) Conditional Coverage Testing

Condition Coverage Testing is a white box testing technique, where the boolean expression is tested for both True and False cases.

89) Compliance Testing

Compliance Testing is performed to audit whether the given application is developed according to the required regulations and standards.

90) Conformance Testing

Conformance Testing is also known as Compliance Testing, which is performed to audit whether the given application is developed according to the required regulations and standards.

91) Regulation Testing

Regulation Testing is also known as Compliance Testing, which is performed to audit whether the given application is developed according to the required regulations and standards.

92) Standards Testing

Standard Testing is also known as Compliance Testing, which is performed to audit whether the given application is developed according to the required regulations and standards.

93) Concurrency Testing

Concurrency Testing is performed to find the defects which result when multiple users are simultaneously using the application.

94) Conversion Testing

Conversion Testing is performed to verify the working of the programs which can convert the data from one format to another required data format.

95) Confirmation Testing

Confirmation Testing is also known as Re-testing, which is performed on the newer version of the application to verify whether the defects reported earlier are fixed by the developer.

96) Consistency Testing

Consistency Testing is performed to check the consistency in UI, APIs, Documentation etc, even-though they are not specified in the requirement specifications.

97) Decision Coverage Testing

Decision Coverage Testing is also known as Branch Coverage Testing, where the testers will make sure that all the possible program branches resulting from each decision point are tested at-least once.

98) Destructive Testing

Destructive Software Testing is performed on the application with an intention of failing it, to test how robust the application is by testing the unpredictable user behavioral with an intention of finding the failures.

99) Dependency Testing

Dependency Testing is performed to examine the application's requirements, initial states, configuration of an existing Software to find the defects or imperfections that are stopping the application from reaching the desired functionality.

100) Domain Testing

Domain Testing is performed on the applications with a minimum selected number of inputs, for testing the output of the system and for also verifying that the application doesn't accept any invalid inputs.

101) Data Warehouse Testing

Data Warehouse Testing is also known as ETL Testing.

ETL (Extract Transform Load) is used in the Data Warehousing projects for performing the below:

1. Extracting the data from different data sources
2. Transforming or applying calculations on the extracted data as per the destination requirement
3. Loads the transformed data into a Data Warehouse.

ETL Testing is performed to check whether the data that is loaded from the source to the destination is accurate and correct.

102) Data Flow Testing

Data Flow Testing falls into white box and structural testing category, which tests the paths found in the program control flow by exploring the variables data receiving and variable data usage program flows.

103) Documentation Testing

Documentation Testing tests the documented artifacts that are created before, during and after the testing of the application.

104) Deployment Testing

Testing the application before, during and after its deployment to avoid any production issue is known as Deployment Testing.

105) Desk Checking

Desk Checking is performed by the developers in a manual way to test whether the underlying algorithm of a program in a line by line basis with an intention of identifying bugs.

106) Error Handling Testing

Error Handling Testing is performed on the application to check whether the application is capable of handling the probable errors.

107) Equivalence Partition Testing

Equivalence Partition Testing is a Software Testing Technique (also known as Equivalence Class Partitioning), which derives the limited test cases with an intention of reducing time, by partitioning the possible input test data into different partitions and then picking at-least one test data from each partition for deriving test cases.

108) Fault Injection Testing

Fault Injection Testing is performed by intentionally introducing the faults into the code and then testing to increase the test coverage.

109) Formal Verification Testing

Formal Verification Testing is performed using mathematical techniques to check whether the application is build according to the requirement specifications.

110) Verification Testing

Verification Testing is performed to check whether the application is developed according to the given requirement specifications.

111) Validation Testing

Validation Testing is performed to check whether the application has met the end user needs.

112) Happy Path Testing

Happy Path Testing is performed on the application with know inputs which result expected output.

113) Golden Path Testing

Golden Path Testing is also known as Happy Path Testing, which is performed with know inputs which result expected output.

114) Sunny Day Testing

Sunny Day Testing is also known as Happy Path Testing, which is performed with know inputs which result expected output.

115) Keyword Driven Testing

Keyword Driven Testing is a test automation framework used by non-technical, which uses keywords in a spreadsheet for calling reusable functions in the automation framework.

116) Incremental Integration Testing

Incremental Integration Testing is performed in Integration Testing phase which comes after Unit Testing, where the developers integrate the individual units/modules one by one in an incremental way with an intention of finding the integration level bugs.

117) Big Bang Integration Testing

Big Bang Integration Testing is performed in Integration Testing phase which comes after Unit Testing, where the developers integrate all the individual units/modules at a go with an intention of finding the integration level bugs.

118) Bottom Up Integration Testing

Bottom Up Integration Testing is performed in Integration Testing phase which comes after Unit Testing, where the developers integrate the lower level individual units/modules with their higher level units/modules until all units/modules are covered.

119) Hybrid Integration Testing

Hybrid Integration Testing is a combination of Top Down Integration and Bottom Up Integration approaches.

120) System Integration Testing

System Integration Testing is performed on the complete system which consists of many sub-systems, to check how the individual modules/units interact with each when integrated into a complete system. The main purpose of this testing is to check the interactions between the system components and is tested when the new module is added to the complete system.

121) Top Down Integration Testing

Top Down Integration Testing is performed in Integration Testing phase which comes after Unit Testing, where the developers integrate the top level individual units/modules with their lower level units/modules until all units/modules are covered.

122) Bi-directional Integration Testing

Bi-directional Integration Testing is also known as Hybrid Integration Testing, which is a combination of Top Down Integration and Bottom Up Integration approaches.

123) Sandwich Integration Testing

Sandwich Testing is also known as Hybrid Integration Testing, which is a combination of Top Down Integration and Bottom Up Integration approaches.

124) Interface Testing

Interface Testing is performed to test the communication between two different software systems or components.

125) Inter-System Testing

When a single application is hosted across different locations, the process of verifying whether the correct and centralized data is flowing across each location is known as Inter-System Testing.

126) Iterative Testing

Iterative Testing is a process of updating the application with small changes and testing them to allow the application evolve over time with smaller changes rather than larger changes.

127) Interoperability Testing

Interoperability Testing is performed to check whether the application is able to communicate with other applications or components without any compatibility issue.

128) Loop Testing

Loop Testing is one of the white box testing techniques used for testing the loops/iterative statements inside the application programs.

129) Logic Coverage Testing

Testing the logical internal structure of the application code based on some logical expressions which are formalized in mathematics is known as Logic Coverage Testing

130) Work flow Testing

Testing the application like being used by its end user, by deriving the business process work flows is known as Work Flow Testing.

131) Web Testing

Testing the web applications with an intention of identifying bugs is known as Web Testing.

132) Web Application Testing

Web Application Testing is also known as Web Testing, which is nothing but testing the web applications with an intention of identifying bugs.

133) All Pairs Testing

All Pairs Testing is performed with an intention of achieving full test coverage, where test cases are derived for the application for testing all different combinations of parameters (i.e. input fields – radio buttons text fields etc., configurations/settings, variables etc.)

134) Pairwise Testing

Pairwise Testing is also known as All Pairs Testing, which is performed with an intention of achieving full test coverage, where test cases are derived for the application for testing all different combinations of parameters (i.e. input fields – radio buttons text fields etc., configurations/settings, variables etc.)

135) Assertion Testing

Adding Assertions (i.e. Boolean Expressions) to the programming code, which will fail when there are any issues identified at the point where these assertions are inserted into the code is known as Assertion Testing.

136) Age Testing

The process of checking the sustainability, durability and performance ability of the application to work in the future is known as Age Testing.

137) Active Testing

Testers directly interacting with the application and testing it is known as Active Testing.

138) Accessibility Testing

Testing whether the application can be used by the users having different disabilities (i.e. hearing disability, blindness etc.) is known as Accessibility Testing.

139) Audit Testing

Audit Testing is the process of auditing the testing process with an intention of evaluating it.

140) Aesthetics Testing

Applying the creativity, passion and visual feel (a kind of art with passion) while creating or using the different things like Testing Artifacts, Test Data, Test Automation Code etc. to increase the sense of readability and appeal to our senses (reading, feeling etc.) is known as Aesthetics Testing.

141) Hypothesis Based Testing

Instead of using some testing methodologies which will help us in identifying the defects in the application, Hypothesis Based Testing is a defect-centric approach where the goal is to identify the probable defects in the application directly identified without using any

testing methodologies/artifacts/process etc. And also, based on the defects identified, the testing methodology, artifacts and tools are created/improved for further effective testing.

142) Upgrade Testing

Testing whether the older version of the application are working properly after upgrading them to latest versions along with testing the upgrade process is known as Upgrade Testing.

143) Thread Testing

Testing performed on the key functional capabilities of a specific thread during the early stages of the Integration Testing phase is known as Thread Testing.

144) Testability Testing

Testability Testing is the process of finding the effort level for performing different testing activities.

145) Statement Coverage Testing

Statement Testing is a white box testing technique, which is used as part of measuring the code coverage by checking whether each and every statement of the application is executed at-least once as part of testing process.

146) Line Coverage Testing

Line Coverage Testing is also known as Statement Coverage Testing, is a white box testing technique, which is used as part of measuring the code coverage by checking whether each and every statement of the application is executed at-least once as part of testing process.

147) Segment Coverage Testing

Segment Coverage Testing is also known as Statement Coverage Testing, is a white box testing technique, which is used as part of measuring the code coverage by checking whether each and every statement of the application is executed at-least once as part of testing process.

148) Stability Testing

Stability Testing is performed to check whether the application is robust enough to work continuously for a long period of time with failing or crashing.

149) Storage Testing

Storage Testing is performed to check whether the application is storing the proper data correctly and has sufficient disk space to store the data to avoid any unexpected termination.

150) Structural Testing

Structural Testing is also known as White Box Testing, which tests the internal code and program structure of the software.

151) Scenario Testing

Scenario Testing is performed by deriving and using the end user real world scenarios, which involves all the detailed steps for testing end-to-end working of the software.

152) Scripted Testing

In Scripted Testing, the tester blindly uses the pre-designed set of test cases and don't use his cognitive ability for testing the application.

153) Unscripted Testing

In Unscripted Testing, the tester uses his cognitive ability for generating different ideas and choosing the appropriate testing methodologies on the fly for testing the application.

154) Specifications Based Testing

Specifications Based Testing is also known as Black Box Testing, which is performed to test the visual functioning of the software without looking into its internal code structure, design and implementation.

155) Input Output Driven Testing

Input Output Driven Testing is also known as Black Box Testing, which is performed to test the visual functioning of the software by providing the input data without looking into its internal code structure, design and implementation.

156) Requirements Based Testing

Requirements Based Testing is also known as Black Box Testing, which is performed to test the visual functioning of the software by using the requirement specifications without looking into its internal code structure, design and implementation.

157) State Transitioning Testing

State Transitioning Testing is performed to test the various state transitions that the application goes through on providing the different input conditions.

158) Recovery Testing

Recovery Testing is performed by intentionally making an application to crash or make its hardware to fail, to check how quickly the application is able to recover to its normal state after application crash or hardware failure.

159) Ramp Testing

Ramp Testing is type of Endurance Testing, which is performed by raising an input continuously until the application crashes/breakdown.

160) Risk Based Testing

Risk Based Testing is performed by prioritizing the tests based on their risk and testing them. (i.e. Test cases which result in high risk on failure will be tested first).

161) Qualification Testing

Qualification Testing is a systematic check performed to test whether the applications qualifies its requirement specifications.

162) Passive Testing

Passive Testing is an opposite of Active Testing, where the application is tested without any manual interaction, by just observation and monitoring it.

163) Parallel Testing

Parallel Testing is performed by testing the multiple applications or multiple modules of the same application simultaneously.

164) Pilot Testing

Pilot Testing is a mock test performed before the application hits production, by testing the major functionality of the application as part of rehearsal by installing it in the user simulated environment.

165) Protocol Testing

Protocol Testing is performed in the communication industry domains, by testing the working of the different communication protocols used in the applications.

166) Operation Testing

Operation Testing is performed to verify the operational readiness of the application by mimicking the production environment, before releasing into the production.

167) Orthogonal Array Testing

Orthogonal Array Testing is performed on the application having huge input data, by creating the test cases using the Orthogonal Arrays technique which produces huge input data combinations for testing.

168) OATS Testing

OATS Testing stands for Orthogonal Array Testing, which is performed on the application having huge input data, by creating the test cases using the Orthogonal Arrays technique which produces huge input data combinations for testing.

169) Model Based Testing

Model Based Testing is performed by deriving the test cases by using any model (i.e. Decision Table Model, State Transition Model etc.)

170) Modularity Driven Testing

Modularity Driven Testing is nothing but an Automation Testing Framework, where the test automation scripts are created separately for different individual modules in the application.

171) Migration Testing

Migration Testing is performed to test whether the old legacy application is successfully migrated to a new application without any data and functional/non-functional problems.

Hope you have found this article useful.

Kindly provide your valuable feedback which will motivate us to write more this kind of articles.

If you come across any other Software Testing Types which are not covered in this article, feel free to comment in this article and we will update the article accordingly.

