

### Introduction

There are numerous types of system testing that can be done as part of the system testing and deliver process. Below is a list of test types identified by Guru 99. This is probably the most comprehensive set of tests that I have ever seen, and it makes a great guide when planning your quality assurance activities. Naturally, the team selects that group of tests that make the most sense in the circumstances.

### 1-25

- 1. Acceptance Testing:** Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. This testing process is usually performed by customer representatives.
- 2. Accessibility Testing:** Type of testing which determines the usability of a product to the people having disabilities (deaf, blind, mentally disabled, etc.). The evaluation process should be conducted by persons with disabilities.
- 3. Active Testing:** Type of testing consisting in introducing test data and analyzing the execution results. This testing process is usually conducted by the testing teams.
- 4. Agile Testing:** A software testing practice that follows the principles of the agile manifesto, emphasizing testing from the perspective of customers who will utilize the system. This type of testing is usually performed by QA teams.
- 5. Age Testing:** Type of testing which evaluates a system's ability to perform in the future. The evaluation process is conducted by testing teams.
- 6. Ad-hoc Testing:** Testing performed without planning and documentation – the tester tries to 'break' the system by randomly trying the system's functionality. This testing process is performed by testing teams.
- 7. Alpha Testing:** Type of testing a software product or system conducted at the developer's site. This testing process is usually performed by the end user.
- 8. Assertion Testing:** Type of testing consisting in verifying if the conditions confirm the product requirements. This type of testing is performed by the testing teams.
- 9. API Testing:** Testing technique similar to unit testing in that it targets the code level. API testing differs from unit testing in that it is typically a QA task and not a developer task.
- 10. All-pairs Testing:** Combinatorial testing method that tests all possible discrete combinations of input parameters. This testing process is usually performed by the testing teams.

### 1-25 (cont)

- 11. Automated Testing:** Testing technique that uses automation testing tools to control the environment set-up, test execution and results reporting. This testing process is performed by a computer and is used inside the testing teams.
- 12. Basis Path Testing:** A testing mechanism which derives a logical complexity measure of a procedural design and is used this as a guide for defining a basic set of execution paths. This testing process is normally used by testing teams when defining test cases.
- 13. Backward Compatibility Testing:** A testing method which verifies the behavior of the developed software with older versions of the test environment. This testing process is performed by testing teams.
- 14. Beta Testing:** Final testing before releasing application for commercial purpose. This testing process is typically done by end-users or others.
- 15. Benchmark Testing:** Testing technique that uses representative sets of programs and data designed to evaluate the performance of computer hardware and software in a given configuration. This testing process is performed by testing teams.
- 16. Big Bang Integration Testing:** A testing technique which integrates individual program modules only when everything is ready. This testing process is performed by the testing teams.
- 17. Binary Portability Testing:** Technique that tests an executable application for portability across system platforms and environments, usually for conformation to an ABI specification. This testing process is performed by the testing teams.
- 18. Boundary Value Testing:** Software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams.
- 19. Bottom Up Integration Testing:** In bottom up integration testing, module at the lowest level are developed first and other modules which go towards the 'main' program are integrated and tested one at a time. This testing process is usually performed by the testing teams.
- 20. Branch Testing:** Testing technique in which all branches in the program source code are tested at least once. This is done by the developer.
- 21. Breadth Testing:** A test suite that exercises the full functionality of a product but does not test features in detail. This testing process is performed by testing teams.
- 22. Black box Testing:** A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality. This testing process is performed by QA teams.

### 1-25 (cont)

**23. Code-driven Testing:** Testing technique that uses testing frameworks (such as x Unit) that allow the execution of unit tests to determine whether various sections of the code are acting as expected under various circumstances. This testing process is performed by the development teams.

**24. Compatibility Testing:** Testing technique that validates how well a software performs in a particular hardware/software/operating system/network environment. This testing process is performed by the testing teams.

**25. Comparison Testing:** Testing technique which compares the product strengths and weaknesses with previous versions or other similar products. Can be performed by tester, developers, product managers or product owners.

### 26-49

**26. Component Testing:** Testing technique similar to unit testing but with a higher level of integration – testing is done in the context of the application instead of just directly testing a specific method. Can be performed by testing or development teams.

**27. Configuration Testing:** A testing technique which determines minimal and optimal configuration of hardware and software, and the effect of adding or modifying resources such as memory, disk drives and CPU. This testing process is performed by the performance testing engineers.

**28. Condition Coverage Testing:** Type of software testing where each condition is executed by making it true and false, in each of the ways at least once. This testing process is typically made by the automation testing teams.

**29. Compliance Testing:** Type of testing which checks whether the system was developed in accordance with standards, procedures and guidelines. This testing process is usually performed by external companies which offer “Certified OGC Compliant” brand.

**30. Concurrency Testing:** Multi-user testing geared towards determining the effects of accessing the same application code, module or database records. This testing process is usually done by performance engineers.

**31. Conformance Testing:** The process of testing that an implementation conforms to the specification on which it is based. This testing process is usually performed by testing teams.

**32. Context Driven Testing:** An Agile Testing technique that advocates continuous and creative evaluation of testing opportunities in light of the potential information revealed and the value of that information to the organization at a specific moment. This testing process is usually performed by Agile testing teams.

### 26-49 (cont)

**33. Conversion Testing:** Testing of programs or procedures used to convert data from existing systems for use in replacement systems. This testing process is usually performed by the QA teams.

**34. Decision Coverage Testing:** Type of software testing where each condition/decision is executed by setting it on true/false. This testing process is typically made by the automation testing teams.

**35. Destructive Testing:** Type of testing in which the tests are carried out to the specimen's failure, in order to understand a specimen's structural performance or material behavior under different loads. This testing process is usually performed by QA teams.

**36. Dependency Testing:** Testing type which examines an application's requirements for pre-existing software, initial states and configuration in order to maintain proper functionality. It is usually performed by testing teams.

**37. Dynamic Testing:** Term used in software engineering to describe the testing of the dynamic behavior of code. This testing process is typically performed by testing teams.

**38. Domain Testing:** White box testing technique which contains checks that the program accepts only valid input. This testing process is usually done by software development teams and occasionally by automation testing teams.

**39. Error-Handling Testing:** Software testing type which determines the ability of the system to properly process erroneous transactions. This testing process is usually performed by the testing teams.

**40. End-to-end Testing:** Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate. This testing process is performed by QA teams.

**41. Endurance Testing:** Type of testing which checks for memory leaks or other problems that may occur with prolonged execution. This testing process is usually performed by performance engineers.

**42. Exploratory Testing:** Black box testing technique performed without planning and documentation. This testing process is usually performed by manual testers.

**43. Equivalence Partitioning Testing:** Software testing technique that divides the input data of a software unit into partitions of data from which test cases can be derived. This testing process is usually performed by the QA teams.

### 26-49 (cont)

- 44. Fault injection Testing:** Element of a comprehensive test strategy that enables the tester to concentrate on the manner in which the application under test is able to handle exceptions. This testing process is performed by QA teams.
- 45. Formal verification Testing:** The act of proving or disproving the correctness of intended algorithms underlying a system with respect to a certain formal specification or property, using formal methods of mathematics. This testing process is usually performed by QA teams.
- 46. Functional Testing:** Type of black box testing that bases its test cases on the specifications of the software component under test. This testing process is performed by testing teams.
- 47. Fuzz Testing:** Software testing technique that provides invalid, unexpected, or random data to the inputs of a program – a special area of mutation testing. Fuzz testing is performed by testing teams.
- 48. Gorilla Testing:** Software testing technique which focuses on heavily testing of one particular module. This testing process is performed by quality assurance teams, usually when running full testing.
- 49. Gray Box Testing:** A combination of Black Box and White Box testing methodologies:\*\* testing a piece of software against its specification but using some knowledge of its internal workings. This testing process can be performed by either development or testing teams.

### 50-105

- 50. Glass box Testing:** Similar to white box testing, based on knowledge of the internal logic of an application's code. This testing process is performed by development teams.
- 51. GUI software Testing:** The process of testing a product that uses a graphical user interface, to ensure it meets its written specifications. This is normally done by the testing teams.
- 52. Globalization Testing:** Testing method that checks proper functionality of the product with any of the culture/locale settings using every type of international input possible. This testing process is performed by the testing team.
- 53. Hybrid Integration Testing:** Testing technique which combines top-down and bottom-up integration techniques in order leverage benefits of these kind of testing. This testing process is usually performed by the testing teams.
- 54. Integration Testing:** The phase in software testing in which individual software modules are combined and tested as a group. This testing process is usually conducted by testing teams.

### 50-105 (cont)

- 55. Interface Testing:** Testing conducted to evaluate whether systems or components pass data and control correctly to one another. This testing process is usually performed by both testing and development teams.
- 56. Install/uninstall Testing:** Quality assurance work that focuses on what customers will need to do to install and set up the new software successfully. It may involve full, partial or upgrades install/uninstall processes and is typically done by a software testing engineer in conjunction with the configuration manager.
- 57. Internationalization Testing:** The process which ensures that product's functionality is not broken and all the messages are properly externalized when used in different languages and locale. This testing process is usually performed by the testing teams.
- 58. Inter-Systems Testing:** Testing technique that focuses on testing the application to ensure that interconnection between application functions correctly. This testing process is usually done by the testing teams.
- 59. Keyword-driven Testing:** Also known as table-driven testing or action-word testing, is a software testing methodology for automated testing that separates the test creation process into two distinct stages:\*\* a Planning Stage and an Implementation Stage. It can be used by either manual or automation testing teams.
- 60. Load Testing:** Testing technique that puts demand on a system or device and measures its response. It is usually conducted by the performance engineers.
- 61. Localization Testing:** Part of software testing process focused on adapting a globalized application to a particular culture/locale. This testing process is normally done by the testing teams.
- 62. Loop Testing:** A white box testing technique that exercises program loops. This testing process is performed by the development teams.
- 63. Manual Scripted Testing:** Testing method in which the test cases are designed and reviewed by the team before executing it. It is done by manual testing teams.
- 64. Manual-Support Testing:** Testing technique that involves testing of all the functions performed by the people while preparing the data and using these data from automated system. This testing process is conducted by testing teams.
- 65. Model-Based Testing:** The application of Model based design for designing and executing the necessary artifacts to perform software testing. This testing process is usually performed by testing teams.

### 50-105 (cont)

**66. Mutation Testing:** Method of software testing which involves modifying programs' source code or byte code in small ways in order to test sections of the code that are seldom or never accessed during normal tests execution. This testing process is normally conducted by testers.

**67. Modularity-driven Testing:** A software testing technique which requires the creation of small, independent scripts that represent modules, sections, and functions of the application under test. This testing process is usually performed by the testing team.

**68. Non-functional Testing:** Testing technique which focuses on testing of a software application for its non-functional requirements. Can be conducted by the performance engineers or by manual testing teams.

**69. Negative Testing:** Also known as "test to fail" – testing method where the tests' aim is showing that a component or system does not work. This testing process is performed by manual or automation testers.

**70. Operational Testing:** Testing technique conducted to evaluate a system or component in its operational environment. This testing process is performed by testing teams.

**71. Orthogonal array Testing:** Systematic, statistical way of testing which can be applied in user interface testing, system testing, regression testing, configuration testing and performance testing. This testing process is performed by the testing team.

**72. Pair Testing:** Software development technique in which two team members work together at one keyboard to test the software application. One does the testing and the other analyzes or reviews the testing. This can be done between one Tester and Developer or Business Analyst or between two testers with both participants taking turns at driving the keyboard.

**73. Passive Testing:** Testing technique consisting in monitoring the results of a running system without introducing any special test data. This testing process is performed by the testing team.

**74. Parallel Testing:** A testing technique which has the purpose to ensure that a new application which has replaced its older version has been installed and is running correctly. This testing process is conducted by the testing team.

**75. Path Testing:** Typical white box testing which has the goal to satisfy coverage criteria for each logical path through the program. This testing process is usually performed by the development team.

**76. Penetration Testing:** A testing method which evaluates the security of a computer system or network by simulating an attack from a malicious source. This testing process is conducted by specialized penetration testing companies.

### 50-105 (cont)

**77. Performance Testing:** Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. This testing process is usually conducted by the performance engineer.

**78. Qualification Testing:** Testing against the specifications of the previous release, usually conducted by the developer for the consumer, to demonstrate that the software meets its specified requirements.

**79. Ramp Testing:** Type of testing consisting in raising an input signal continuously until the system breaks down. This testing process may be conducted by the testing team or the performance engineer.

**80. Regression Testing:** Type of software testing that seeks to uncover software errors after changes to the program (e.g., bug fixes or new functionality) have been made, by retesting the program. This testing process is performed by the testing teams.

**81. Recovery Testing:** Testing technique which evaluates how well a system recovers from crashes, hardware failures, or other catastrophic problems. This testing process is performed by the testing teams.

**82. Requirements Testing:** Testing technique which validates that the requirements are correct, complete, unambiguous, and logically consistent and allows designing a necessary and sufficient set of test cases from those requirements. This testing process is performed by QA teams.

**83. Security Testing:** A process to determine that an information system protects data and maintains functionality as intended. It can be performed by testing teams or by specialized security-testing companies.

**84. Sanity Testing:** A testing technique which determines if a new software version is performing well enough to accept it for a major testing effort. This testing process is performed by the testing teams.

**85. Scenario Testing:** Testing activity that uses scenarios based on a hypothetical story to help a person think through a complex problem or system for a testing environment. This testing process is performed by the testing teams.

**86. Scalability Testing:** Part of the battery of non-functional tests which tests a software application for measuring its capability to scale up – be it the user load supported, the number of transactions, the data volume, etc. This testing process is conducted by the performance engineer.



### 50-105 (cont)

- 87. Statement Testing:** A white box testing which satisfies the criterion that each statement in a program is executed at least once during program testing. It is usually performed by the development team.
- 88. Static Testing:** A form of software testing where the software isn't actually used it checks mainly for the sanity of the code, algorithm, or document. It is used by the developer who wrote the code.
- 89. Stability Testing:** Testing technique which attempts to determine if an application will crash. This testing process is usually conducted by the performance engineer.
- 90. Smoke Testing:** Testing technique which examines all the basic components of a software system to ensure that they work properly. Typically, smoke testing is conducted by the testing team, immediately after a software build is made.
- 91. Storage Testing:** Testing type that verifies the program under test stores data files in the correct directories and that it reserves sufficient space to prevent unexpected termination resulting from lack of space. This testing process is usually performed by the testing team.
- 92. Stress Testing:** A testing technique which evaluates a system or component at or beyond the limits of its specified requirements. This testing process is usually conducted by the performance engineer.
- 93. Structural Testing:** White box testing technique which takes into account the internal structure of a system or component and ensures that each program statement performs its intended function. This testing process is usually performed by the software developers.
- 94. System Testing:** The process of testing an integrated hardware and software system to verify that the system meets its specified requirements. This testing process is conducted by the testing teams in both development and target environment.
- 95. System integration Testing:** Testing process that exercises a software system's coexistence with others. This testing process is usually performed by the testing teams.
- 96. Top Down Integration Testing:** Testing technique that involves starting at the stop of a system hierarchy at the user interface and using stubs to test from the top down until the entire system has been implemented. This testing process is conducted by the testing teams.
- 97. Thread Testing:** A variation of top-down testing technique where the progressive integration of components follows the implementation of subsets of the requirements. This testing process is usually performed by the testing teams.

### 50-105 (cont)

- 98. Upgrade Testing:** Testing technique that verifies if assets created with older versions can be used properly and that user's learning is not challenged. This testing process is performed by the testing teams.
- 99. Unit Testing:** Software verification and validation method in which a programmer tests if individual units of source code are fit for use. This testing process is usually conducted by the development team.
- 100. User Interface Testing:** A type of testing which is performed to check how user-friendly the application is. This testing process is performed by testing teams.
- 101. Usability Testing:** Testing technique which verifies the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. This testing process is usually performed by end users.
- 102. Volume Testing:** Testing which confirms that any values that may become large over time (such as accumulated counts, logs, and data files), can be accommodated by the program and will not cause the program to stop working or degrade its operation in any manner. This testing process is usually conducted by the performance engineer.
- 103. Vulnerability Testing:** A type of testing which regards application security and has the purpose to prevent problems which may affect the application integrity and stability. This testing process can be performed by the internal testing teams or outsourced to specialized companies.
- 104. White box Testing:** Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. This testing process is performed by software developers.
- 105. Workflow Testing:** A scripted end-to-end testing technique which duplicates specific workflows which are expected to be utilized by the end-user. This testing process is usually conducted by testing teams.