**TYPES OF TESTING**

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. It is usually performed by the customer.
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 is conducted by persons having disabilities.  
   What is Software Testing Why Testing is Important
3. **Ad-hoc Testing:**Testing performed without planning and documentation - the tester tries to 'break' the system by randomly trying the system's functionality. It is performed by the testing team.   
   **Difference between Ad-hoc & Exploratory testing**
4. **Alpha Testing:**Type of testing a software product or system conducted at the developer's site. Alpha Testing is performed by the tester within the Organisation.  
   **Beta Testing  
   Difference between Alpha & Beta testing**
5. **Assertion Testing:**Type of testing consisting in verifying if the conditions confirm the product requirements. It is performed by the testing team.
6. **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.
7. **Automated Testing:**Testing technique that uses Automation Testing tools to control the environment set-up, test execution and results reporting. It is performed by a computer and is used inside the testing teams.
8. **Backward Compatibility Testing:**Testing method which verifies the behavior of the developed software with older versions of the test environment. It is performed by testing team.
9. **Beta Testing:**Type of testing a software product or system conducted at the Client’s site. Beta Testing is performed by the real end users.
10. **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.
11. **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. It is usually performed by the testing teams.
12. **Branch Testing:**Testing technique in which all branches in the program source code are tested at least once. This is done by the developer.
13. **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. It is performed by QA teams.
14. **Code-driven Testing:**Testing technique that uses testing frameworks (such as xUnit) that allow the execution of unit tests to determine whether various sections of the code are acting as expected under various circumstances. It is performed by the development teams.
15. **Compatibility Testing:**Testing technique that validates how well a software performs in a particular hardware/software/operating system/network environment. It is performed by the testing teams.
16. **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.
17. **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. It is typically made by the Automation Testing teams.
18. **Decision Coverage Testing:**Type of software testing where each condition/decision is executed by setting it on true/false. It is typically made by the automation testing teams/developers.
19. **Dynamic Testing:**Term used in software engineering to describe the testing of the dynamic behavior of code. It is typically performed by testing teams.
20. **Error-Handling Testing:**Software testing type which determines the ability of the system to properly process erroneous transactions. It is usually performed by the testing teams.
21. **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. It is performed by QA teams.
22. **Exploratory Testing:**Black box testing technique performed without planning and documentation. It is usually performed by manual testers.
23. **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. it is usually performed by the QA teams.
24. **Functional Testing:**Type of black box testing that bases its test cases on the specifications of the software component under test. It is performed by testing teams.
25. **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.
26. **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. It can be performed by either development or testing teams.
27. **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.
28. **Hybrid Integration Testing:**Testing technique which combines top-down and bottom-up integration techniques in order leverage benefits of these kind of testing. It is usually performed by the testing teams.
29. **Integration Testing:**The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.
30. **Interface Testing:**Testing conducted to evaluate whether systems or components pass data and control correctly to one another. It is usually performed by both testing and development teams.
31. **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 the software testing engineer in conjunction with the configuration manager.
32. **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.
33. **Load Testing:**Testing technique that puts demand on a system or device and measures its response. It is usually conducted by the performance engineers.
34. **Modularity-driven Testing:**Software testing technique which requires the creation of small, independent scripts that represent modules, sections, and functions of the application under test. It is usually performed by the testing team.
35. **Non-functional Testing:**Testing technique which focuses on testing of a software application for its non-functional requirements. Can be conducted by testing teams.
36. **Negative Testing:**Also known as "test to fail" - testing method where the tests' aim is showing that a component or system does not work. It is performed by manual or automation testers.
37. **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.
38. **Parallel Testing:**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. It is conducted by the testing team.
39. **Penetration Testing:**Testing method which evaluates the security of a computer system or network by simulating an attack from a malicious source. Usually they are conducted by specialized penetration testing companies.
40. **Performance Testing:**Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.
41. **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. It is performed by the testing teams.
42. **Retesting**: Type of software testing which is focused on testing the issue again after the reported issue has been resolved.
43. **Sanity Testing:**Testing technique which determines if a new software version is performing well enough to accept it for a major testing effort. It is performed by the testing teams.
44. **Statement Testing:**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.
45. **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.
46. **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.
47. **Stress Testing:**Testing technique which evaluates a system or component at or beyond the limits of its specified requirements. It is usually conducted by the performance engineer.
48. **System Testing:**The process of testing an integrated hardware and software system to verify that the system meets its specified requirements. It is conducted by the testing teams in both development and target environment.
49. **Top Down Integration Testing:**Testing technique that involves starting at the top of a system hierarchy at the user interface and using stubs to test from the top down until the entire system has been implemented. It is conducted by the testing teams.

**52. Unit Testing:**Software verification and validation method in which a programmer tests if individual units of source code are fit for use. It is usually conducted by the development team.

**53. 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. It is performed by software developers.

**NOTE: HEADINGS HIGHLIGHTED IN BLUE ARE EITHER VERY LESS TALKED ABOUT/OUT OF CONCERN IN TERMS OF BLACK BOX TESTING**