

Acceptance criteria - Exit criteria that a component or system or application must satisfy in order to be accepted by an end user or customer or other authorized entity.

Acceptance Testing: Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

Accessibility Testing: Verifying a product is accessible to the people having disabilities (deaf, blind, mentally disabled etc.).

Accuracy - Capability of software product or application to provide the agreed results or effects with specified degree of precision.

Actual Result - Behavior produced or observed when a component or system is tested.

Ad Hoc Testing: A testing phase where the tester tries to break the system by randomly trying the system's functionality. Can include negative testing as well. See also Monkey Testing.

Adaptability - is the capability of software product to be adapted for specified multiple environments without

applying actions other than those provided for the specific purpose for the software considered.

Agile Testing: Testing practice for projects using agile methodologies, treating development as the customer of testing and emphasizing a test-first design paradigm. See also Test Driven Development.

Automated Software Quality (ASQ): The use of software tools, such as automated testing tools, to improve software quality.

Alpha Testing - Operational testing by intended users / customers or an independent test team at the developer's location. Alpha testing is often employed for off the shelf software as a form of internal acceptance testing.

Analyzability- capability of the software product or system to be diagnosed for deficiencies or causes of failures in the software or for the parts to be modified to be identified.

Automated Testware - Test ware used in test automation such as tool scripts etc.

Back to Back Testing - Software Testing in which more than one variants of a software system are executed

with the same inputs and the outputs compared & analyzed in cases of discrepancies.

Basis Path Testing: A white box test case design technique that uses the algorithmic flow of the program to design tests.

Basis Set: The set of tests derived using basis path testing.

Beta Testing: Testing of a re-release of a software product conducted by customers.

Benchmark Test - A test that is be used to compare components or systems to each other or to a standard.

Big Bang Testing: Type of integration testing in which software elements and hardware elements are combined all at once.

Binary Portability Testing: Testing an executable application for portability across system platforms and environments, usually for conformation to an ABI specification.

Black Box Testing: Testing based on an analysis of the specification of a piece of software without reference to its internal workings. The goal is to test how well the

component conforms to the published requirements for the component.

Blocked test case - A test case which cannot be executed because the pre-conditions for its execution are not fulfilled.

Bottom Up Testing: An approach to integration testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

Boundary Testing: Test which focus on the boundary or limit conditions of the software being tested. (Some of these tests are stress tests).

Boundary Value - Input value or output value which is on the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. The example can be minimum or maximum value of a range.

Boundary Value Coverage - %age of boundary values that have been exercised by a test suite.

Branch Coverage - %age of branches that have been exercised by a test suite. 100% branch coverage implies

both 100% decision coverage and 100% statement coverage.

Bug: A fault in a program which causes the program to perform in an unintended or unanticipated manner.

Boundary Value Analysis: BVA is similar to Equivalence Partitioning but focuses on "corner cases" or values that are usually out of range as defined by the specification. This means that if a function expects all values in range of negative 100 to positive 1000, test inputs would include negative 101 and positive 1001.

Branch Testing: Testing in which all branches in the program source code are tested at least once.

Breadth Testing: A test suite that exercises the full functionality of a product but does not test features in detail.

Business Process Based Testing - An approach to software testing in which test cases are designed based on descriptions and knowledge of business processes and components.

CAST: Computer Aided Software Testing.

Capture/Replay Tool: A test tool that records test input as it is sent to the software under test. The input cases stored can then be used to reproduce the test at a later time. Most commonly applied to GUI test tools.

CMM: The Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes.

Cause Effect Graph: A graphical representation of inputs and the associated outputs effects which can be used to design test cases.

Changeability - capability of the software or system to enable specified modifications to be implemented.

Classification Tree Method - Black box test design technique in which test cases, described by means of a classification tree which are designed to execute combinations of representatives of input and output domains.

Code Complete: Phase of development where functionality is implemented in entirety; bug fixes are all that are left. All functions found in the Functional Specifications have been implemented.

Code Coverage: An analysis method that determines which parts of the software have been executed (covered) by the test case suite and which parts have not been executed and therefore may require additional attention.

Code Inspection: A formal testing technique where the programmer reviews source code with a group who ask questions analyzing the program logic, analyzing the code with respect to a checklist of historically common programming errors, and analyzing its compliance with coding standards.

Code Walkthrough: A formal testing technique where source code is traced by a group with a small set of test cases, while the state of program variables is manually monitored, to analyze the programmer's logic and assumptions.

Compatibility Testing: Testing whether software is compatible with other elements of a system with which it should operate, e.g. browsers, Operating Systems, or hardware.

Compliance - Capability of the software product to adhere to standards or regulations in laws and similar prescriptions.

Compliance Testing - Process of testing to determine the compliance of component or system.

Component - minimal software item that can be tested in isolation.

Component Integration Testing - Software Testing performed to expose defects in the interfaces & interaction between components.

Compound Condition Testing - Testing tTwo or more single conditions joined by means of a logical operator.

Concurrency Testing: Multi-user testing geared towards determining the effects of accessing the same application code, module or database records. Identifies and measures the level of locking, deadlocking and use of single-threaded code and locking semaphores.

Conformance Testing: The process of testing that an implementation conforms to the specification on which it is based. Usually applied to testing conformance to a formal standard.

Context Driven Testing: The context-driven school of software testing is flavor of Agile Testing 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 right now.

Conversion Testing: Testing of programs or procedures used to convert data from existing systems for use in replacement systems.

Cyclomatic Complexity: A measure of the logical complexity of an algorithm, used in white-box testing.

Daily Build - It is a development activity where a complete system is compiled and linked every day so that a consistent system is available at any time including all latest changes.

Data Driven Testing: Testing in which the action of a test case is parameterized by externally defined data values, maintained as a file or spreadsheet. A common technique in Automated Testing.

Data Flow Testing - White box test design technique in which software test cases are designed to execute definition and use pairs of variables.

Database Integrity Testing: Testing the methods used to access and manage the data or database to ensure

that during access to the database, data is not corrupted or unexpectedly deleted, updated or created.

Decision Condition Testing - White box test design technique in which test cases are designed to execute condition outcomes and decision outcomes.

Decision Table Testing - Black box test design techniques in which test cases are designed to execute the combinations of inputs put in a decision table.

Defect: Nonconformance to requirements or functional / program specification

Defect Management - Process of recognizing, investigating, taking action and disposing of defects. It involves recording and tracking of defects by classifying them and identifying the impact.

Defect Masking - Occurrence in which one defect prevents the detection of another.

Dependency Testing: Examines an application's requirements for pre-existing software, initial states and configuration in order to maintain proper functionality.

Depth Testing: A test that exercises a feature of a product in full detail.

Design Based Testing - Approach to test in which test cases are designed based on architecture and detailed design of the system.

Development Testing - A Formal or informal testing conducted by developers during the implementation of a component or system.

Documentation Testing - Testing of the quality of the documentation.

Dynamic Testing: Testing software through executing it. See also Static Testing.

Efficiency Testing - Process of testing to determine the efficiency of a software product.

Elementary Comparison Testing - Black box test design technique in which the concept of condition determination coverage is used to design the test cases are to execute combinations of inputs.

Endurance Testing: Checks for memory leaks or other problems that may occur with prolonged execution.

End-to-End testing: Testing 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.

Equivalence Class: A portion of a component's input or output domains for which the component's behavior is assumed to be the same from the component's specification.

Equivalence Partitioning: A test case design technique for a component in which test cases are designed to execute representatives from equivalence classes.

Error Guessing - Test design technique where the experience of a tester is used to identify what defects might be present in the component or system under test.

Error seeding - Process of intentionally adding known defects to a component or system for the purpose of monitoring the rate of detection, removal and estimating the number of remaining defects.

Exhaustive Testing: Testing which covers all combinations of input values and preconditions for an element of the software under test.

Exploratory Testing: It is an informal test design technique where a tester actively controls the design of

the tests as those tests are performed while testing to design new and better tests.

Functional Decomposition: A technique used during planning, analysis and design; creates a functional hierarchy for the software.

Functional Specification: A document that describes in detail the characteristics of the product with regard to its intended features.

Functional Testing:

- Testing the features and operational behavior of a product to ensure they correspond to its specifications.
- Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

Frozen Test Basis - Test basis document that can only be modified or amended by a formal change control process.

Gorilla Testing: Testing one particular module, functionality heavily.

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.

High Order Tests: Black-box tests conducted once the software has been integrated.

Independent Test Group (ITG): A group of people whose primary responsibility is software testing.

Inspection: A group review quality improvement process for written material. It consists of two aspects; product (document itself) improvement and process improvement (of both document production and inspection).

Integration Testing: Testing of combined parts of an application to determine if they function together correctly. Usually performed after unit and functional testing. This type of testing is especially relevant to client/server and distributed systems.

Installation Testing: Confirms that the application under test recovers from expected or unexpected events without loss of data or functionality. Events can include shortage of disk space, unexpected loss of communication, or power out conditions.

Localization Testing: This term refers to making software specifically designed for a specific locality.

Loop Testing: A white box testing technique that exercises program loops.

Monkey Testing: Testing a system or an Application on the fly, i.e just few tests here and there to ensure the system or an application does not crash out.

Negative Testing: Testing aimed at showing software does not work. Also known as "test to fail".

Path Testing: Testing in which all paths in the program source code are tested at least once.

Penetration testing: Evaluating the security of a computer system or network by hacking. it is the process of actively evaluating your information security measures. For more details, please see my other post on Penetration Testing

Performance Testing: Testing conducted to evaluate the compliance of a system or component with specified performance requirements. Often this is performed using an automated test tool to simulate large number of users.

Positive Testing: Testing aimed at showing software works. Also known as "test to pass".

Quality Assurance: All those planned or systematic actions necessary to provide adequate confidence that a product or service is of the type and quality needed and expected by the customer.

Quality Audit: A systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives.

Quality Circle: A group of individuals with related interests that meet at regular intervals to consider problems or other matters related to the quality of outputs of a process and to the correction of problems or to the improvement of quality.

Quality Control: The operational techniques and the activities used to fulfill and verify requirements of quality.

Ramp Testing: Continuously raising an input signal until the system breaks down.

Recovery Testing: Confirms that the program recovers from expected or unexpected events without loss of data or functionality. Events can include shortage of disk space, unexpected loss of communication, or power out conditions.

Regression Testing: Retesting a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made.

Release Candidate: A pre-release version, which contains the desired functionality of the final version, but which needs to be tested for bugs (which ideally should be removed before the final version is released).

Sanity Testing: Brief test of major functional elements of a piece of software to determine if its basically operational.

Scalability Testing: Performance testing focused on ensuring the application under test gracefully handles increases in work load.

Security Testing: Testing which confirms that the program can restrict access to authorized personnel and that the authorized personnel can access the functions available to their security level.

Smoke Testing: A quick-and-dirty test that the major functions of a piece of software work. Originated in the hardware testing practice of turning on a new piece of hardware for the first time and considering it a success if it does not catch on fire.

Soak Testing: Running a system at high load for a prolonged period of time. For example, running several times more transactions in an entire day (or night) than would be expected in a busy day, to identify and performance problems that appear after a large number of transactions have been executed.

Software Testing: A set of activities conducted with the intent of finding errors in software.

Static Analysis: Analysis of a program carried out without executing the program.

Static Analyzer: A tool that carries out static analysis.

Static Testing: Analysis of a program carried out without executing the program.

Storage Testing: Testing 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 is external storage as opposed to internal storage.

Stress Testing: Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. Often this is performance testing using a very high level of simulated load.

Structural Testing: Testing based on an analysis of internal workings and structure of a piece of software.

System Testing: Testing that attempts to discover defects that are properties of the entire system rather than of its individual components.

Testability: The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

Testing:

- The process of exercising software to verify that it satisfies specified requirements and to detect errors.
- The process of analyzing a software item to detect the differences between existing and required conditions

(that is, bugs), and to evaluate the features of the software item (Ref. IEEE Std 829).

- The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.

Test Bed: An execution environment configured for testing. May consist of specific hardware, OS, network topology, configuration of the product under test, other application or system software, etc. The Test Plan for a project should enumerated the test beds(s) to be used.

Test Case:

- Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc.
- A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Driven Development: Testing methodology associated with Agile Programming in which every chunk of code is covered by unit tests, which must all pass all the time, in an effort to eliminate unit-level and regression bugs during development. Practitioners of TDD write a lot of tests, i.e. an equal number of lines of test code to the size of the production code.

Test Driver: A program or test tool used to execute a tests. Also known as a Test Harness.

Test Environment: The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers.

Test First Design: Test-first design is one of the mandatory practices of Extreme Programming (XP).It requires that programmers do not write any production code until they have first written a unit test.

Test Harness: A program or test tool used to execute a tests.

Test Plan: A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing

tasks, who will do each task, and any risks requiring contingency planning.

Test Procedure: A document providing detailed instructions for the execution of one or more test cases.

Test Script: Commonly used to refer to the instructions for a particular test that will be carried out by an automated test tool.

Test Specification: A document specifying the test approach for a software feature or combination of features and the inputs, predicted results and execution conditions for the associated tests.

Test Suite: A collection of tests used to validate the behavior of a product. The scope of a Test Suite varies from organization to organization. There may be several Test Suites for a particular product for example. In most cases however a Test Suite is a high level concept, grouping together hundreds or thousands of tests related by what they are intended to test.

Test Tools: Computer programs used in the testing of a system, a component of the system, or its documentation.

Thread Testing: A variation of top-down testing where the progressive integration of components follows the implementation of subsets of the requirements, as opposed to the integration of components by successively lower levels.

Top Down Testing: An approach to integration testing where the component at the top of the component hierarchy is tested first, with lower level components being simulated by stubs. Tested components are then used to test lower level components. The process is repeated until the lowest level components have been tested.

Total Quality Management: A company commitment to develop a process that achieves high quality product and customer satisfaction.

Traceability Matrix: A document showing the relationship between Test Requirements and Test Cases.

Usability Testing: Testing the ease with which users can learn and use a product.

Use Case: The specification of tests that are conducted from the end-user perspective. Use cases tend to focus

on operating software as an end-user would conduct their day-to-day activities.

Unit Testing: Testing of individual software components.

Validation: The process of evaluating software at the end of the software development process to ensure compliance with software requirements. The techniques for validation is testing, inspection and reviewing.

Verification: The process of determining whether or not the products of a given phase of the software development cycle meet the implementation steps and can be traced to the incoming objectives established during the previous phase. The techniques for verification are testing, inspection and reviewing.

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.

Walkthrough: A review of requirements, designs or code characterized by the author of the material under review guiding the progression of the review.

White Box Testing: Testing based on an analysis of internal workings and structure of a piece of software. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

Workflow Testing: Scripted end-to-end testing which duplicates specific workflows which are expected to be utilized by the end-user.