Software Testing Methods and Techniques

TAXONOMY SHEET

It is very important to differentiate between software testing and software quality assurance. Software testing is a too late stage if quality is not built into the software product.

Software Quality Assurance

Software testing is a popular risk management strategy. It is used to verify that functional requirements were met.

Software Testing

Quality Control

Quality control is defined as the processes and methods used to monitor work and observe whether requirements are met. It focuses on reviews and removal of defects before shipment of products

Software Configuration Management

Software configuration management is concerned with labeling, tracking, and controlling changes in the software elements of a system. It controls the evolution of a software system by managing versions of its software components and their relationships

Black-Box Testing (Functional)

With black-box testing, the tester views the program as a black-box and is completely unconcerned with the internal structure of the program or system.

White-Box Testing (Structural)

The tester examines the internal structure of the pro-gram or system. Test data is driven by examining the logic of the program or system, without concern for the program or system requirements.

Gray-Box Testing (Functional and Structural)

Gray-box testing is a combination of black- and white-box testing. The tester studies the requirements specifications and communicates with the developer to understand the internal structure of the system.

60 TESTING TECHNIQUES COVERING DIFFERENT PURPOSES ALONG THEIR CLASSIFICATION

Technique\ Description	Manual	Automated	Static	Dynamic	Black-box	White-box
Acceptance Testing Final testing based on the end-user/customer specifications, or based on use by end-users/customers over a defined period of time	★	★		★	★	
Ad-hoc Testing Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it	★				★	
Alpha Testing Testing of an application when development is nearing completion; minor design changes may still be made as a result of such testing. Typically done by end-users or others, not by programmers or testers	★			★	★	
Basis Path testing Identifying tests based on flow and paths of a program or system		★		★		★
Beta Testing Testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end-users or others, not by programmers or testers	★			★	★	
Black-box testing Testing cases generated based on the system's functionality		★		★	★	
Bottom-up testing Integrating modules or programs starting from the bottom		★		★		★
Boundary value testing Testing cases generated from boundary values of equivalence classes		★		★	★	
Branch coverage testing Verifying that each branch has true and false outcomes at least once		★		★		★
Branch/condition coverage testing Verifying that each condition in a decision takes on all possible outcomes at least once		★		★		★
Cause–effect graphing Mapping multiple simultaneous inputs that may affect others to identify their conditions to test		★		★	★	
Comparison testing Comparing software weaknesses and strengths to competing products	★	★		★	★	★

60 TESTING TECHNIQUES COVERING DIFFERENT PURPOSES ALONG THEIR CLASSIFICATION

2

Technique\ Description	Manual	Automated	Static	Dynamic	Black-box	White-box
Compatibility testing Testing how well software performs in a particular hardware/software/operating system/network environment	★	★				★
Condition coverage testing Verifying that each condition in a decision takes on all possible outcomes at least once		★		★		★
CRUD testing Building a CRUD matrix and testing all object creations, reads, updates, and deletions		★		★	★	
Database testing Checking the integrity of database field values		★		★		★
Decision tables Table showing the decision criteria and the respective actions		★		★	★	
Desk checking Developer reviews code for accuracy	★			★		★
End-to-end Testing Similar to system testing; the “macro” end of the test scale; 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	★	★			★	
Equivalence partitioning Each input condition is partitioned into two or more groups. Test cases are generated from representative valid and invalid classes		★		★		
Exception testing Identifying error messages and exception handling processes and conditions that trigger them		★		★	★	
Exploratory testing Often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it	★			★	★	
Free form testing Ad hoc or brainstorming using intuition to define test cases		★		★	★	
Gray-box testing A combination of black-box and white-box testing to take advantage of both		★		★	★	★
Histograms A graphical representation of measured values organized according to the frequency of occurrence used to pinpoint hot spots	★				★	
Incremental integration testing Continuous testing of an application as new functionality is added; requires that various aspects of an application’s functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed; done by programmers or by testers	★	★		★	★	
Inspections Formal peer review that uses checklists, entry criteria, and exit criteria	★		★		★	★
Integration testing Testing of combined parts of an application to determine if they function together correctly. The “parts” can be code modules, individual applications, or client and server applications on a network. This type of testing is especially relevant to client/server and distributed systems.	★	★		★	★	
JADs Technique that brings users and developers together to jointly design systems in facilitated sessions	★				★	★
Load testing Testing an application under heavy loads, such as testing of a Web site under a range of loads to determine at what point the system’s response time degrades or fails	★	★		★		★
Mutation testing A method for determining if a set of test data or test cases is useful, by deliberately introducing various code changes (“bugs”) and retesting with the original test data/cases to determine if the “bugs” are detected. Proper implementation requires large computational resources.	★	★		★	★	

60 TESTING TECHNIQUES COVERING DIFFERENT PURPOSES ALONG THEIR CLASSIFICATION

3

Technique\ Description	Manual	Automated	Static	Dynamic	Black-box	White-box
Orthogonal array testing Mathematical technique to determine which variations of parameters need to be tested	★		★		★	
Pareto analysis Analyze defect patterns to identify causes and sources	★				★	
Performance testing Term often used interchangeably with “stress” and “load” testing. Ideally “performance” testing (and any other “type” of testing) is defined in requirements documentation or QA or Test Plans	★	★		★	★	★
Positive and negative testing Testing the positive and negative values for all inputs		★		★	★	
Prior defect history testing Test cases are created or rerun for every defect found in prior tests of the system	★		★		★	
Prototyping General approach to gather data from users by building and demonstrating to them some part of a potential application		★		★	★	
Random testing Technique involving random selection from a specific set of input values where any value is as likely as any other		★		★	★	
Range testing For each input identifies the range over which the system behavior should be the same		★		★	★	
Recovery testing Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems	★	★		★		★
Regression testing Testing a system in light of changes made during a development spiral, debugging, maintenance, or the development of a new release				★	★	
Risk-based testing Measures the degree of business risk in a system to improve testing	★		★		★	
Run charts A graphical representation of how a quality characteristic varies with time	★		★		★	
Sandwich testing Integrating modules or programs from the top and bottom simultaneously	★		★		★	★
Sanity testing Typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort. For example, if the new software is crashing systems every five minutes, bogging down systems to a crawl, or destroying databases, the software may not be in a “sane” enough condition to warrant further testing in its current state	★	★		★	★	
Security testing Testing how well the system protects against unauthorized internal or external access, willful damage, etc.; may require sophisticated testing techniques	★	★				★
State transition testing Technique in which the states of a system are first identified and then test cases are written to test the triggers to cause a transition from one condition to another state		★		★	★	
Statement coverage testing Every statement in a program is executed at least once		★		★		★
Statistical profile testing Statistical techniques are used to develop a usage profile of the system that helps define transaction paths, conditions, functions, and data tables	★		★		★	
Stress testing Term often used interchangeably with “load” and “performance” testing. Also used to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, or large complex queries to a database system	★	★		★		

60 TESTING TECHNIQUES COVERING DIFFERENT PURPOSES ALONG THEIR CLASSIFICATION

4

Technique\ Description	Manual	Automated	Static	Dynamic	Black-box	White-box
Structured walkthroughs A technique for conducting a meeting at which project participants examine a work product for errors	★			★	★	★
Syntax testing Data-driven technique to test combinations of input syntax		★	★	★	★	
System testing Black-box type testing that is based on overall requirements specifications; covers all combined parts of a system	★	★		★	★	
Table testing Testing access, security, and data integrity of table entries		★		★		★
Thread testing Combining individual units into threads of functionality which together accomplish a function or set of functions		★		★		★
Top-down testing Integrating modules or programs starting from the top		★		★	★	★
Unit testing The most “micro” scale of testing; to test particular functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses	★	★	★			★
Usability testing Testing for “user-friendliness.” Clearly this is subjective, and will depend on the targeted end-user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers	★	★		★	★	
User acceptance testing Determining if software is satisfactory to an end-user or customer	★	★		★	★	
White-box testing Test cases are defined by examining the logic paths of a system		★		★		★

V-Model

Testing Lifecycle Review

Development Phases Vs. Testing Types