DEVELOPING TEST CASES

What is a Test Case?

Although most people believe that everyone understands the difference between test plans, test procedures, and test cases, experience has shown that this is not the case. There is great deal of confusion among project management, test engineering, and software development of what these are, their intent, and their value.

Test Cases are derived from Test Scenarios that are identified in a Test Procedure. A Test Procedure is a logical unit of test engineering activity. It could be testing of a service, a use case, a function, an interface, or a component. Within the Test Procedure are 1 or more Test Scenarios. These Test Scenarios can be at a unit level, integration, end to end, or even system. Finally, within each Test Scenario are 1 or more Test Cases.

A Test Case has a stimulus with a set of conditions and an expected result. A Test Case may be complex with multiple conditions and results or a simple Pass or Fail. The level of complexity is left to the test designer and the expected skill level of the person who will execute the tests.

A Test Case has an expected result. When the Test Case is executed, the actual result is compared to the expected result. A Test Case is repeatable.

How to develop a Test Case?

There are several types of test cases. The most obvious set are positive or “happy path” test cases. These are test cases associated with correct and expected inputs, behaviors, and results. Requirements-based testing is an example of this.

Negative tests address the need to test the robustness of the software and the system to unexpected or varying inputs, states, or conditions. Test cases for negative testing may not always be explicitly stated in a Test Procedure but it is always implied and expected to be done by the Test Engineer.

Ad-hoc or seat-of-the-pants testing is touted as an effective methodology for testing. This is neither good test engineering nor an effective use of test resources. An alternative to ad-hoc testing is exploratory testing. This type of testing works well when software is handed to a test engineer in lieu of documentation. It also can work when legacy code is involved.

Additionally, test cases may be developed for abnormal events, stress, reliability, performance, benchmark, loading, and loss of connections, resources, and services.

Various techniques are used when designing good test cases. The following chart lists some of the applicable testing techniques that can be used.

	Type of Testing											
Type of Test Case	**Installation & Configuration Testing**	**Requirements Testing**	**Services Testing**	**Performance and Stress Testing**	**Business Use Case Testing**	**Regression Testing**	**User Acceptance Testing**	**Interoperability**	**Compatibility**	**Benchmark**	**Compliance & Conformance**	**Reliability & Longevity**
Nominal values			X			X						
Bounds checking						X						
Negative tests			X		X							
Pathing			X			X						
Sequencing	X		X		X	X	X					
Events			X	X								
Normal & abnormal terminations	X		X	X	X	X	X					X
Error Messages	X		X		X	X	X					
Error Logging	X		X		X	X	X					
Initialization of services	X		X		X		X		X			
Availability of Services	X		X	X	X		X		X			X
Integration of services			X	X	X		X	X				X
Verification of each requirement		X								X	X	
White Box		X	X	X								
Black Box	X	X	X	X	X	X	X	X		X		X

Utilizing a Test Case

Each test case identified in a Test Procedure should be executed at lease once during a formal test cycle. Depending on a number of factors, a test case may be run multiple times during a product release. These factors could include effectiveness of the test case, defect fixes associated with the test case, new features, or inclusion of the test case in a regression suite of tests.

Executing a test case may involve multiple steps beyond a simple pass or fail. A test case may initially fail, be fixed, and then fail again based on a new underlying issue uncovered by the fix (the so called “onion” effect).