Software Testing - Test Cases

What are test cases in software testing, how are they designed and why they are so important to the entire testing scenario? Read on to know more.

Testing an application's or program's working is not black-and-white. A program may work in one situation or condition, but may fail in another. It is up to the software tester, to ensure that a program works correctly in all possible conditions. Consider the example given below.

Imagine a program which adds two numbers. The program must accept two numerical inputs, perform the addition and display the output. But certain conditions exist, which can hamper the functioning of the program. Like if one input is zero. The program should correctly display the output, which is the number itself. What if one number is negative? Then the program should perform subtraction and correctly assign a positive or negative sign to the answer.

The above example is a very simple program. With larger programs written in different languages, there are more conditions and each one is significantly more complex. The software tester must identify said conditions and test the program's working in each case. The program is said to be working effectively, only when it has handled all such conditions in the correct manner. No condition should end the program abruptly. Managing such conditions and presenting them in a clear and concise format, is the reason for writing test cases.

What is a Test Case?

Simply put, a test case is a scenario made up of a sequence of steps and conditions or variables, where test inputs are provided and the program is run using those inputs, to see how it performs. An expected result is outlined and the actual result is compared to it. Certain working conditions are also present in the test case, to see how the program handles the conditions.

Every requirement or objective that the program is expected to achieve, needs at least one test case. Realistically, it definitely takes more than one test case to determine the true functionality of the application being tested. The mechanism used to judge the result of the test case, i.e. whether the program has failed or passed the test, is called a test oracle.

Test cases, at root level, are used to measure how a program handles errors or tricky situations such as if one input is incorrect or if both inputs are incorrect. They are also expected to expose hidden logical errors in the program's code, that have gone undetected.

Typical Structure of a Test case

A formal written test case can be divided into three main parts:

Information
Information consists of general information about the test case such as a case identifier, case creator info, test case version, formal name of the test case, purpose or brief description of the test case and test case dependencies. It should also include specific hardware and software requirements (if any) and setup or configuration requirements.

Activities
This part consists of the actual test case activities such as the environment that should exist during testing, activities to be done at the initialization of the test, activities to be done after test case is performed, step-by-step actions to be done while testing and the input data that is to be supplied for testing.

Results
Results are the outcomes of a performed test case. Result data consists of information about expected results, which is the criteria necessary for the program to pass the test and the actual recorded results.

Test Case Format

Two sample formats for writing test cases are:

Detailed

Test		***Test***	***Test***		***Test***	***Test***	***Expected***	***Actual***	***Verdict:***	

Case Id	*Purpose*	*Created By*	*Environment*	*Prerequisites*	*Procedure*	*Data*	*Result*	*Result*	*Pass/Fail*	*Comments*
Serial no assigned to test case	Brief idea about case	Name of test creator	Software or hardware in which the test case is executed	Conditions that should be fulfilled before the test is performed	Steps to be performed in test	Inputs, variables and data	What the program should do	What is actually done	Status of the test	Notes on the procedure

Simple

Step No.	Serial no of step
Step or Activity	Detailed operation or procedure
Criteria for Success	Expected result
Status	Whether the code passed the test or not

Designing test cases can be time-consuming in a testing schedule, but they are worth the time spent because they can prevent unnecessary retesting or debugging or at least lower the rate of such operations. Organizations can take the test case approach in their own context and according to their own perspectives. Some follow a general approach while others may opt for a more detailed and complex approach. It is important for you to decide between the two extremes and decide what would work best for you.

By Amol Vyavhare
Last Updated: 3/26/2012

Related Articles

Software Testing Methodologies

Software Testing - How To Go About For Beginners

Software Testing - White Box Testing Strategy

Software Testing - Black Box Testing Strategy

Software Verification & Validation Model - An Introduction

Software Testing Interview Questions

Software Testing - Check Lists For Software Tester

Software Testing Interview Questions and Answers

Software Testing - Bug and Statuses Used During A Bug Life Cycle

Manual Testing Interview Questions

Post Comment | View Comments

Your Comments: