

1. Overview

The Big Picture

All software problems can be termed as bugs. A software bug usually occurs when the software does not do what it is intended to do or does something that it is not intended to do. Flaws in specifications, design, code or other reasons can cause these bugs. Identifying and fixing bugs in the early stages of the software is very important as the cost of fixing bugs grows over time. So, the goal of a software tester is to find bugs and find them as early as possible and make sure they are fixed.

Testing is context-based and risk-driven. It requires a methodical and disciplined approach to finding bugs. A good software tester needs to build credibility and possess the attitude to be explorative, troubleshooting, relentless, creative, diplomatic and persuasive.

As against the perception that testing starts only after the completion of coding phase, it actually begins even before the first line of code can be written. In the life cycle of the conventional software product, testing begins at the stage when the specifications are written, i.e. from testing the product specifications or product spec. Finding bugs at this stage can save huge amounts of time and money.

Once the specifications are well understood, you are required to design and execute the test cases. Selecting the appropriate technique that reduces the number of tests that cover a feature is one of the most important things that you need to take into consideration while designing these test cases. Test cases need to be designed to cover all aspects of the software, i.e. security, database, functionality (critical and general) and the user interface. Bugs originate when the test cases are executed.

As a tester you might have to perform testing under different circumstances, i.e. the application could be in the initial stages or undergoing rapid changes, you have less than enough time to test, the product might be developed using a life cycle model that does not support much of formal testing or retesting. Further, testing using different operating systems, browsers and the configurations are to be taken care of.

Reporting a bug may be the most important and sometimes the most difficult task that you as a software tester will perform. By using various tools and clearly communicating to the developer, you can ensure that the bugs you find are fixed.

Using automated tools to execute tests, run scripts and tracking bugs improves efficiency and effectiveness of your tests. Also, keeping pace with the latest developments in the field will augment your career as a software test engineer.

What is software? Why should it be tested?

Software is a series of instructions for the computer that perform a particular task, called a program; the two major categories of software are system software and application software. System software is made up of control programs. Application software is any program that processes data for the user (spreadsheet, word processor, payroll, etc.).

A software product should only be released after it has gone through a proper process of development, testing and bug fixing. Testing looks at areas such as performance, stability and error handling by setting up test scenarios under controlled conditions and assessing the results. This is why exactly any software has to be tested. It is important to note that software is mainly tested to see that it meets the customers' needs and that it conforms to the standards. It is a usual norm that software is considered of good quality if it meets the user requirements.

What is Quality? How important is it?

Quality can briefly be defined as "a degree of excellence". High quality software usually conforms to the user requirements. A customer's idea of quality may cover a breadth of features - conformance to specifications, good performance on platform(s)/configurations, completely meets operational requirements (even if not specified!), compatibility to all the end-user equipment, no negative impact on existing end-user base at introduction time.

Quality software saves good amount of time and money. Because software will have fewer defects, this saves time during testing and maintenance phases. Greater reliability contributes to an immeasurable increase in customer satisfaction as well as lower maintenance costs. Because maintenance represents a large portion of all software costs, the overall cost of the project will most likely be lower than similar projects.

Following are two cases that demonstrate the importance of software quality:

Ariane 5 crash June 4, 1996

- Maiden flight of the European Ariane 5 launcher crashed about 40 seconds after takeoff
- Loss was about half a billion dollars
- Explosion was the result of a software error
- Uncaught exception due to floating-point error: conversion from a 64-bit integer to a 16-bit signed integer applied to a larger than expected number

- Module was re-used without proper testing from Ariane 4
 - Error was not supposed to happen with Ariane 4
 - No exception handler
- Mars Climate Orbiter - September 23, 1999
- Mars Climate Orbiter, disappeared as it began to orbit Mars.
 - Cost about \$US 125-million
 - Failure due to error in a transfer of information between a team in Colorado and a team in California
 - One team used English units (e.g., inches, feet and pounds) while the other used metric units for a key spacecraft operation.

What exactly does a software tester do? ✓

2 JOBS PUBLICATIONS FOR TESTER QA POSITION

HW1 GOOGLE CLASSROOM
DUE DATE: THURSDAY 11PM