 SOFTWARE TESTING - WHAT IS IT ANYWAY?  
  
Software testing is a way to ensure the quality of software. All types of software require testing, and there is a formal way or method used to test software of every kind.  The formal methods of testing require knowledge of first how software is made or developed.  Testing follows this pattern very closely, each step of the development process is followed by some sort of testing.  Testing not only involves checking the code, but also the various documents that are the end product for each step in the development process. Enter the test case which is a sequence of steps and description of each individual test including what the good situation looks like and what the actual result looks like. For an actual software there can be hundreds of these test cases. Each test case is designed by a software tester or QA Analyst or Software Test Engineer.

THE HOOD

Look who’s come to the neighborhood. Software QA is fun and exciting, it attracts some of the nicest people and conversation around how to get something done right. Have you ever gone to a website and tried some highly useful function like clicking a link to a video and the video doesn’t load or trying to buy a product and can’t get your chosen product to load in the shopping cart. Uggh. Well all of these problems can be prevented by employing highly effective and cool QA talent. What are the effective roles in a software organization? Starting with the QA department, there is the QA Manager, the guru, who is surrounded by QA Analysts, Test Engineers, Automation Engineers, and the great manager also employs outstanding technologies to assist in the testing endeavor – automated testing tools and test job scheduling frameworks, test management systems, bug reporting and tracking software. And you gotta have dollars in the hood, right. The average salary of QA Analyst is $75-80K per year. Automation Engineers make even more $90-95K, and QA Managers a cool $120-140K per year. Not bad, yeah.

ROLES AND RESPONSIBILITIES

Many times as QA Analyst you will be responsible for communicating project status to department heads and even the Corporate Information Officer CIO, so the gift for gab is on the top of the qualifications list, both verbally and in writing. As an Analyst your responsibilities include writing test cases from the requirements, analyzing requirements, and design documents to ensure that the direction of the project favors testability. You will even be responsible for writing and producing a test plan and an overall test strategy for the application and may execute tests manually or by automation with automation software testing tools.

The Test Engineer is responsible for the quality of a product as a whole. They are responsible for the functional performance system and serviceability of the entire software product and performs many of the functions of a QA Analyst, but their role is expanded to the full life cycle of the product and every scope of testing, whereas the Analysts scope of testing would be limited to functional/regression testing for instance.

The Automation Engineer is responsible for developing and executing automated tests for an entire software product. They may convert manual test cases to automation or build automation test cases from scratch. Either way they are responsible for maintenance of the automated test library and the execution of the tests. Test execution may be supervised or unsupervised. For unsupervised testing the automation engineer must diagnose and resolve any technical issues arising from problems with the test harness and making sure the results of pass and fail for each test case run is correct as well as ensuring the artifacts (proof of the errors such as screenshots, videos, etc. ) from the test case are in proper order. The automation engineer builds a software application robot to mimic the user operating the software and gathers data on the running of the application based on data fed into the software where data is needed. Data repositories are kept corresponding to scenarios that the automation engineer designs. These scenarios are the bases for what are called test cases. The automated system stores the result of each test run and reports those results to the team.

The QA Manager acts as a supervisor for the analysts and engineers in a quality organization. The QA manager often roles up their sleeves and performs software testing themselves. They have expanded responsibility in performing applied research of software testing practices to make certain the test processes are running smoothly. They confer with customers to understand any issues in delivered software products and documents those issues. The issues are sent to the respective team members to perform root cause analysis and find the cause behind the issue. The QA Manager is responsible for ensuring that remediation or a resolution is found.

An example of a test case, an example of a bug report.

WHAT IS ISTQB

The ISTQB is an international organization that is all about furthering the software quality assurance trade and ensuring there are reputable certified quality assurance testers available around the world. What can one expect after becoming certified? ISTQB provides a certified career trajectory to those who aspire success in the industry. The CTFL or Certified Tester Foundation Level is the entry level certification in software testing. It ensures that those who are interested in entering the field of software testing understand the core basic fundamental concepts in software testing and how to apply them.

FACTS OF TESTING – Just a taste!

What are the 7 principles of testing?

1. Testing can reveal defects, but cannot reveal that defects do not exist. For instance I can test a shopping cart program and find a defect (bug) in calculating the total order price. But just because I can test this portion of code and see a problem there, does not mean that other problems do not exist elsewhere in the shopping cart system.
2. Exhaustive testing is impossible. Testing everything in a software system is impractical and just cannot be done. The nature of the development process, makes testing everything impossible as software development is continuous and ongoing, always working on a new version or fix. When a bug arises, a new fix to the code may correct the problem, but may also create new bugs, which must be tested for… and the process goes on and on. To make testing finite special methodology is used such as risk analysis (analyzing to determine what parts of the software are the most critical to the customer), using test techniques and prioritization.
3. Early testing saves time and money. The longer a bug sits in the software because other development efforts are taking place and the piece of buggy code may have been used and reused in many other ways and places, makes finding bugs later very expensive to resolve.
4. Defects cluster together. Because defects may be related to a difficult functionality in the software, or due to other defined risks, a large number of defects may pool there. Also defects arising in other areas of the software may have a root cause that leads back to the difficult functionality.
5. Beware of pesticide paradox. After using the same tests over and over they will eventually “wear out” and will not reveal any new defects. To find new bugs, we need new tests and new test data. Just like when you call the exterminator, they come out and spray, but there may be roaches running around in the spray, the exterminator may need to use a new insecticide or different extermination methods.
6. Testing is context dependent. Context is like circumstances. Software is developed for different circumstances and for and in different conditions. Because of this the software must be tested in different ways. For an aircraft navigation software, a harness may be needed for the flight control arm and the structure of the software may be different. The structure of ecommerce shopping cart requires a different testing method or technique than a mobile application. These examples all demonstrate different contexts.
7. Absence-of-errors is a fallacy. When a software is reported to have no errors, it may be because its at the end of the testing cycle, but newly designed tests may reveal more errors. Especially if the new design incorporates different techniques and algorithms for the tests.

WHAT KIND OF MINDSET DOES IT TAKE TO BE A GREAT TESTER?

The best tester sees the endeavor of testing in terms of assisting the developers to reach a higher level of proficiency and quality in their work. This is a positive mindset. They reject the mindset that testing is about criticizing and belittling the author of the work. Yes, the work is critical but the combination of a developer and a tester is a mentee and mentor relationship. The tester does not merely report errors but actively engages with the developers and offers assistance to the developer when needed, either by getting others involved in code reviews or searching for solutions in knowledgebases and the internet. Some of the traits of a good tester include curiosity, a critically eye, attention to detail, motivation for good positive communications and relationships, non-biased and independent judgment.

TEST LEVELS

Component Testing – unit or module testing performed in isolation

Integration Testing – test of multiple modules together including interfaces

System Testing – testing the whole system in a like environment

Acceptance Testing – testing by users to validate software meets user needs in real or like environment.

Alpha testing – at developers site, by the customers

Beta testing – at customers site, by the customers

TEST TYPES

Functional Testing – tests of the functional user requirements, user scenarios

Non-functional Testing – tests of how well system works- usability, performance, security

Black-box Testing – tests software at its external interface (outside)

White-box Testing – tests the internal code characteristics (inside)

Regression Testing – tests for system defects caused by code changes

Purpose: To provide enough information to the student in a rudimentary way so that they will feel competent enough and have motivation to seek certification.

GLOSSARY Glossary.istqb.org

QA Mentor <https://www.youtube.com/watch?v=62jVKgJjRR4>

Make a audio screen video of writing a simple program in eclipse Java and executing test.