Richard Clarke

Is Software Testing Right for Me?

I spent the last couple years testing software and self-learning Android Development. Recently, I felt unsure about which path was best for me. I don't have a mentor at work, and I'm not part of a team of testers, so it's difficult to find where I fit in this field. I decided to research what it means to be a software tester and what the field entails. The more I know about testing, the easier it is to decide my career path.

Software Testing is the field of ensuring high quality software. While it is usually referred to as Quality Assurance, this is a misnomer. According to the Software Testing Fundamentals website, QA refers to the practice of working with engineers, management, and product to optimize processes for creating higher quality software (http://softwaretestingfundamentals.com/). The testing process itself is known as Quality Control.

Quality Control is when testers confirm the functionality of new features and fixes to bugs. Testers confirm new features by coordinating with developers to learn how they work and make sure they function as planned. Additionally, testers find edge cases that might uncover latent problems with a feature. When a software feature doesn't function as expected, it is referred to as a bug.

Customers and internal teams document these bugs. Testers verify them by replicating the steps leading to the bug. The results of these reviews help teams determine weak points in their product. If the bugs do in fact exist, developers are responsible for fixing them. Testers confirm bugs are repaired by developers before the fixes and features are combined into a build.

A build is a collection of the new features and fixes applied to a product. This build is tested in a separate environment before it is released to customers and external users. Testers note areas where the product has been updated and create a test plan; a document testers follow to confirm the quality of the software. Once everything is confirmed, the build overwrites the code in the production environment and is made available to customers. While this might sound like a large amount of work for testers to do on a fixed schedule, there are time saving utilities for this process. Testers write automation scripts for testing the User Interface and Application Programming Interface.

Automation scripts are used to control input of a web browser or device. Typically, testers create these scripts to automatically perform a set of actions on a product. The outcome of these tests determines whether or not a bug has been introduced. For example, if a script is written for clicking a button and the test fails, test engineers review the failure to see what caused it to occur. It could be a myriad of reasons including bugs, loss of connectivity, etc. While UI scripts test the interface, or front end of a product, API tests are used to test the code.

API tests run constantly and ensure the code is free of any bugs that might cause outages or disruptions to the functionality of the software.

Every area I covered reminded me how much I enjoy creating. That's what started my interest in Android development. Nothing beats researching, planning, and putting something together. I don't see this in the field of software testing. There isn't much flexibility in the solutions. There are many different automation tools, but even tests have to conform to a strict set of standards to ensure they run efficiently.

Software testing is interesting, and I'm good at it. But development offers a higher level of flexibility in terms of ways to solve problems. I am glad I researched this field because I'm now certain I want to transition into Android Development.

Sources

Software Quality Assurance. (2018, March 03). Retrieved February 7, 2019, from http://softwaretestingfundamentals.com/software-quality-assurance/