**The Importance of Software Testing**

Gage Biros

CS – 320

February 2nd, 2021

Southern New Hampshire University

Abstract

In the software development life cycle, there are many key development teams that are imperative to the success of any finished product. One of the more important roles that could greatly improve the efficiency of the overall development process is the developer and the tester roles. This paper will be looking at what important habits, or routines to have when working on any kind of software development; as well as what type of testing should be used for each project and how to properly analyze the project requirements to make this adjustment. This will also contain a reflection of an example of the development and testing process and the types of testing that was completed for the project. Here, we will be looking at specific examples of development and testing techniques that were used in this project.

**The Importance of Software Testing**

There are a lot of important roles that are imperative to the entire software development life Cycle (SDLC). Every project is unique, as such, the development process might not be the same as there are different scenarios each project will face. To better understand what the development process will look like, we need to analyze the details like the requirements, and even things like team composition. Every requirement the client gives can play a part on the development process. Software fails for many reasons, but to help find and correct the errors that cause failures we employ testing. There are a couple of different ways we can test software, but knowing the different ways we can test the software can help with the efficiency of the project development. In this example, we developed and utilized a dynamic testing method; JUnit tests help verify and validate the software to ensure the quality of the product. The advantage of JUnit tests in this project helped strengthen development skills, and testing skills which will help not only in this project but in future projects. Also, it helps manage the code more effectively. We are able to write some code, and then develop a JUnit test to validate a certain feature works.

**Summary**

In the project we were working as a software developer for a company and specifically have been asked to develop and test back-end services. Having the clients’ requirements helps understand what exactly we are looking to accomplish. We had three services we needed to develop for the client and each service had its own requirement. I referenced the requirements for each which gave me an idea on how to develop the code, and more importantly, how I should develop the Junit tests to validate and verify that the program is working as intended. All of the testing files will demonstrate how the JUnit tests were written to meet the requirements in detail. If we take a look at the Test Service java file specifically, one technique I used was to confirm that requirements were being met but initializing variables to what should pass the conditions. Once I can run the test and pass the requirements with variables that meet the requirements, then I need to make sure that only valid entries should be possible, so I also created variables that should fail the condition. Our requirements say that the ID name should be no more than 10 characters, same with the name variable, and the description should be no more than 50 characters; in this file I created variables that will pass the conditions and fail the conditions and use methods to call these variables and would prompt the user if the variable pass or fail the requirements of the software. This starts on line 10 in the Task Service test. Actually, my Task Service JUnit test was my best one at an 89 percent total coverage, the others were not that high so I need to evaluate what exactly is causing the coverage to be affected. In the project, what helped me develop the source code effectively was to work on one feature at a time. Once I completed a portion of the project code, I then moved to the JUnit test and developed the test code and developed the test code. In the contact service test, I was able to create a JUnit test within 40 lines of code which tested both valid and invalid entries. I was able to efficiently develop the JUnit test testing all of the conditions for a valid entry.

**Reflection**

In this project, one thing that helped me move along was to develop one feature at a time. It is extremely overwhelming to look at the project as a whole, so it is more manageable to look at it, piece by piece. Then to make sure we are not reproducing errors in the software it is beneficial to also develop JUnit tests one test at a time. This kind of static, white-box testing is best for this situation. In this assignment we are just one developer and tester, companies can save resources by employing developers to test their source code so that by the time it gets to an end user most of the errors can be found and rectified. Here I know the code, and I am the one that will be developing tests for the code, so this is known white-box testing.

In contrast to white-box testing, there is black-box testing where the testers don’t know the structure and analyze the behavior of the program to complete the testing. In this setting we were the developers and the testers as which is more of white -box testing which is said to be more static; however, we also have executed some functionality testing which is more of a black-box, dynamic way of testing the software.

I think a refresher course in java would have gone a long way to the success of my programming in this course. I feel like we were so focused on the testing portion that the actual source code for me was a struggle. One technique I picked up throughout this course was to initialize my testing software that would pass variables, which would allow me to then test certain functions and also verify that all of the requirements were being met, like a two birds one stone mindset. In the Appointment service test code, I set the variables equal to something that should pass the requirements, then utilized the update feature to update the variables to fail the requirements, and if it failed the update because of a restriction of the condition, then I would quickly find out that my update feature works.

When it comes to bias, I cannot understand where I would be okay with developing code, testing it and then passing it as a final product. I think it is an important characteristic as a developer, and a tester, especially one who tests their own code, to have a certain level of integrity that when developing code, we need to have the client’s best interest at heart. Would I consider my product in this close to release to an end user? No. I would however continue to test the source code, and the JUnit tests to ensure I reach a 100% coverage to feel comfortable to release it to the end user.

There is a lot of responsibility as a software developer to ensure the programs we are running are as error free as possible. Some instances are more critical than others, but nevertheless we should hold ourselves to a high standard to help deliver a quality product to the client. I work in property

management and have to analyze the work of many different service providers in the field. If the contracted work isn’t up to par then I make sure I send the company a notice that they won’t get paid until the requirements have been met. Having this experience working on projects in the field, and knowing maybe my software may be used where people’s lives are at risk, then I will have the motivation to develop at a higher standard to push myself to ensure I have left everything I had to put into my work, in my finished product.

**References**

Hambling, Brian Morgan, Peter Samaroo, Angelina Thompson, Geoff Williams, Peter.

(2015). *Software Testing - An ISTQB-BCS Certified Tester Foundation Guide (3rd Edition).* BCS The Chartered Institute for IT. Retrieved from  
https://app.knovel.com/hotlink/toc/id:kpSTAIST01/software-testing-an-istqb/software-testing-an-istqb