Southern New Hampshire University

Project Two

Chantel Williams

CS-320 Software Test Automation

10/12/2022

Project, one required me to code a mobile application and provide unit tests to verify that the application meets the customer’s requirements. In order to satisfy the customer, I need to be sure that my testing approach aligned to the software requirements. Reading through the customer requirements thoroughly was one of the ways I was able to ensure that I followed customer requirements. When I understand what is being asked of me from the customer, I can provide a final product that will satisfy them. When writing my JUnit test, I made sure to open the rubric into a word document, and as I made progress on the assignment, I would highlight each task as I completed it. Using this method allowed me to visually see that I completed each requirement in full.

I wanted to focus on the time it took my code to execute to ensure my code was efficient, because the faster it produces results can help me consider the quality is good. Another way I ensured efficient code was by making sure I had comments in my code, so readers can comprehend what’s going on as well as making sure I named my JUnit test accurate names that coincided with what test was being run at the time. I can also check my overall code quality by using the coverage checker. When using this tool, I was able to see that my overall coverage was above 80%.

Making sure my Junit test followed the customer’s requirements was an easy way to make sure my code was technically sound. For example, when coding for the Task service I needed to be able to update the name and description, so I wrote the following test to be sure that could happen.

Text

Description automatically generated

Since there are many different software testing techniques to use unfortunately, I was unable to use them all. One of the techniques that I was unable to use was white box testing**. “**White Box Testing is a testing technique in which software’s internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security” (Hamilton,2022). I was unable to use this technique because this type of testing involves verifying the functionality of conditional loops and broken or poorly structured paths in the coding processes. I found other ways to ensure my code was free of these errors, so I didn’t necessarily need to use this method.

Dynamic and static testing along with structure-based testing are some software testing techniques that I employed for each milestone. I found that using both techniques were the most effective when working through each milestone. Software development teams use static testing to check their code for errors and bugs without having to execute it. For these milestones I needed to manually test my code so this can was useful because it allowed me to find errors in the early development stages. I was able to use dynamic testing when making my JUnit test for each milestone. “Dynamic Testing is a type of Software Testing which is performed to analyze the dynamic behavior of the code. It includes the testing of the software for the input values and output values that are analyzed” (pp\_pankaj, 2019). I found dynamic testing useful because I was given a list of requirements for my code and to be sure those requirements met the criteria, I had to manually input correct and incorrect information to check the functionally of my code.

 Black box testing is like white box testing, but the difference is that black box testing only “Is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. Only the external design and structure are tested” (mahak\_jain, 2022). White box testing has a lot of practical use because it helps you understand your code while creating test cases to verify functionality. Overall, I think dynamic testing is the most practical because it allows you to manually examine requirements and functionality via handwritten test. This can be time consuming depending on what needs to be tested by it’s a great way to ensure that the required parameters have been validated.

When working on this project I tried to adopt a software developer mindset. It was important to me that I employed caution when testing all working components. It’s also important to appreciate the complexity and interrelationships of the code I was testing because understanding this helps you protect and detect any future bugs. Best practices suggest that you test as often as possible to improve induvial code quality. When writing and testing your own code you can sometimes form a bias and I believe it’s important to stray away from this as much as possible. Some way I tried to limit bias in my own reviews was knowing that my code can always be improved. Throughout these milestones I made sure to take all feedback into consideration in order to develop better code. Being disciplined in your commitment to quality as a software engineering professional is also important because you never want to be put into a position where you are caused to cut corners when writing and testing code. Some ways I plan to avoid this once I'm in my professional career is to always make sure I ask my clients as many questions as possible before starting the project. Another way I will make sure I avoid this is by making sure I provide the proper estimates so that I have enough time to complete each project in full will maintaining good code quality.

**References**

mahak\_jain. (2022, September 15). *Differences between black box testing vs white box testing*. GeeksforGeeks. Retrieved September 30, 2022, from https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/

pp\_pankaj. (2019, May 13). *Software testing: Dynamic testing*. GeeksforGeeks. Retrieved September 11, 2022, from https://www.geeksforgeeks.org/software-testing-dynamic-testing/