**Project Two Submission**

**Summary**

This course has been a difficult one for me. I am not the most experienced coder but I have been learning things as I go. The goal of project one was to create code that aligned with the software requirements and to make sure that the outcome of the test coverage was at an overall high enough percentage for all the individual classes. I was steady with the assignments leading up to the project but lost the proper build when updating early mistakes. After getting feedback from my instructor, I had to do a lot of research on the internet for help in fixing my classes and tests.

Writing the JUnit tests was a challenge for me. The overall quality of the JUnit tests was probably around 50% at first. One thing I could improve on is testing for a scenario that fails. I didn’t realize I was only testing the project for when it worked. I was also supposed to test for failed tests. I figured out how to write a little better test code to run according to the requirements.

**Reflection**

There were different techniques used to perform my software tests. One technique used was unit testing or Junit testing. Unit testing is where blocks of individual code are tested to make sure they work in accordance with the requirements. This type of white-box testing is also done early in the SDLC stages to reduce costs by catching problems early.

There are different types of testing techniques that I did not use for the project. Some of those are acceptance testing, integration testing, systems testing, and static testing. Testing that is done at the end of the project by the final client is called acceptance testing. Acceptance testing tests the software to make sure the final product is ready for release. Before that, integration testing and systems testing occurs. Integration testing ensures that various components work correctly as they are integrated. It happens after the code has passed the necessary unit tests. Systems testing tests the functionality of an entire systems software or an application. This testing normally happens after integrated testing. Static testing happens when a software application is tested without running the code. This could be used to find problems early in writing code.

I am fairly inexperienced when it comes to writing code. The only real practice I have with writing code has been in my courses. The mindset I have been using to complete my work was to try to meet the requirements of an assignment as close as I can. I had a lot of trouble and made some mistakes with different things but I have worked hard and I tried to figure out the problems I encountered. By saying all this, I don’t feel like I have a bias towards the code I wrote. I think I need to improve my coding skills to have my own opinion on the world of coding.

Discipline as a software engineering professional is extremely important for multiple reasons. The quality of a product relies heavily on the level of work put into it. If the development of software is rushed, the quality of the product could turn out low grade. The product could turn out to have all types of bugs that could cause clients to not want to do business with you again. Another problem that could occur is that the software could be released with security vulnerabilities that could be exploited by attackers. A client would want their product to be made fast but not too fast to compromise the integrity of the product. To avoid technical debt as a practitioner in the field, I want to take more time planning out the project I am working on. I need to set higher standards of quality and learn more techniques to better my skills.

**References**

Vijay(March 10, 2024) What is Acceptance Testing (A Complete Guide)

https://www.softwaretestinghelp.com/what-is-acceptance-testing/

JUnit Tutorial | Testing Framework for Java(2024)

https://www.javatpoint.com/junit-tutorial

Static Testing Tutorial(September 27, 2023) What Is Static Testing: Its Features And Best Practices

https://www.lambdatest.com/learning-hub/static-testing