Module 7-2 Project 2

Summary

According to the feedback I would say that the unit testing solutions fit well with the requirements, but the test classes needed to be organized by packages for more effective use of code coverage tools. Using the feedback on my project one submission the file coverage was over 80%. This Shows that JUnit tests used in the code were effective and efficient. I ensured that the code was technically sound by ensuring that the test classes and all other classes aligned with the system requirements and ensured that the test classes covered all possible uses for their classes. For example the testAddTask class where I used data structures and the new task would have the following (“sheogorath”, “get the thing”) to fill in the points in the add task classes and then this ensures a proper test. I tried to keep the code efficient by keeping down the bloat and unnecessary or redundant use of classes and other variables too allow the code to compile quickly and not run through redundant code.

Reflection

I tried my best to use static testing techniques and principles in this project. The reasons for this are that static testing is used to prevent errors later and is an attempt to ensure that all problems are caught early on before the code is executed. The reason I tried to do this is because of my work schedule and limited time I have to spend on the project. I find that this time constraint in real life would fit into a real world example of keeping the cost down from running into errors as the code is executed and would be the most effective and efficient way to test this code. In doing so I kept my overall time involved in testing the project low like you would keep the time involved in a real world example and keeping cost low. I used data structures and a pseudo technical review and walkthrough through the code. The data structures were used in the code and I followed how data or how the code would execute in the system and what would call what and what classes relied on or possible dependencies. The pseudo technical review, which a regular technical review is a meeting that looks at possible defects, I would pick a point during the building of the project to focus on defects and to review these defects against documents online and references online.

In this project and acting as a software tester I tried to be very cautious and tried not to be to experimental or go to far out into possible solutions or ways of coding. I did this to keep possible problems or defects down to a minimum and tried to keep to already known and tested techniques since this project isn’t too complex in its requirements it shouldn’t require complicated solutions. With this keeping the techniques used in the code to very simplistic and well known techniques kept the time invested into testing the code down to a minimum. I tried to limit bias in my review of the code this was also helped by getting the feedback on the code that keeps bias in check. From the software developer side I would believe that bias would be a possible issue in the development of this code since the person testing it would also be the developer but maybe only a minor problem since the developer would want their code to work without error. The only possible problem or bias I could see in this scenario would be overlooking some minor issues in the code to cut down on time spent in the testing methods.

The importance of being disciplined in the software engineering profession would be important to ensuring that quality software is being built and deployed. I can see personally that with my time constraints that cutting corners is all too easy to fall into just save on time and to meet deadlines. In a real world scenario, this could also be a issue since meeting dead lines is important but a developer cannot fall into these traps since this could possibly affect the quality of the code and when cutting corners while testing the code all this can cause is problems happening when the code is deployed and thus lessening the trust in your product and cost since the code would need to be fixed after deployment and a multitude of other problems.