Luis Sanchez
SNHU

Summary:

My unit testing approach aligned with all of the software requirements. For example, it aligned with the Contact, Task, and Appointment requirements of the ID String that cannot be longer than 10 characters, shall not be null and shall not be updatable by using functions to ensure the Junit test flags an error. Following all the resources in each module and feedback from outside sources I knew my Junit tests were of quality. Test coverage for the java files had 80% coverage or higher.

I ensured my code was technically sound by going over the software requirements and using if statements to make sure the requirements are meant. For example, one if statement I used in the Task software was:

```
if (taskId == null || taskId.length() > 10) {
        throw new IllegalArgumentException("Task ID cannot be null and must be at most 10
        characters long");
          }
```

This if statement checks if the Task ID entered is MORE than 10 characters long.
I ensured my code was efficient with the knowledge I have from previous courses and outside sources like youtube and my peers in this class.

Reflection:

Luis Sanchez
SNHU

A software testing technique employed in this project was whitebox testing. A software testing method that involves examining the internal structure and workings of a system or application. The primary objectives of white box testing are to ensure the completeness, correctness, and reliability of the software code.

A software testing technique I did not employ in this project was Automated Testing. It Involves the use of automated testing tools to execute pre-scripted tests on the software. It is beneficial for repetitive tasks, regression testing, and ensuring consistency in testing processes.

White box testing is valuable during the early stages of software development for unit testing and integration testing. It is particularly useful for identifying logical errors, uncovering issues in the code structure, and ensuring that all code paths are executed.

Automated testing is suitable for projects with repetitive tasks, frequent code changes, and a need for quick and consistent test execution. It is particularly valuable in regression testing, performance testing, and cases where a large number of test scenarios need to be executed.

The mindset i adapted for this project was a cautious approach. Caution was paramount due to the recognition that the quality and reliability of the software heavily depend on the effectiveness of the testing process.

In the review of my code, the way to limit bias is by adopting a systematic and objective approach. The assessment of my code was based on the objective criteria such as adherence to coding standards, logical coherence, and requirements of the software application. I think bias is a concern when testing your own code which is why I believe implementing practices such as peer reviews is a great way to overcome that difficulty.

Luis Sanchez
SNHU

There is a great importance of being disciplined as a software engineer professional because it leads to more quality code and less technical debt. It is important to not cut corners when writing and testing code because cutting corners can lead to bugs and future problems down the road when the product/software is published to clients. A way I will avoid technical debt in the future is by following coding standards that ensures consistency and readability. This will help prevent the introduction of unnecessary complexities and makes the codebase more maintainable. For example, consistently using meaningful variable names enhances code clarity.