CS 320 Project Two

1. **Summary**
   1. Describe your unit testing approach for each of the three features.
      1. To what extent was your approach **aligned to the software requirements**? Support your claims with specific evidence.

When we look at the software there are three that come to mind. Its Contact Service, Task Service, and Appointment Service. For Appointment Service I tested it for appointment ID uniqueness, so if its past the 10 character limit ‘testAddAppointmentInvalidID()’. For Task Service the test makes sure that the object task is meeting the requirement and I use ‘testUpdateTaskName()’ to make sure that the character limit is valid. Lastly for Contact Service we test validate the contact creation, updating and etc. We use ‘testDeleteContactByID()’ to make sure that contacts are removed properly. These are some specific examples and now we will talk about how its aligned with the software requirements.

Its align with software requirements by writing Junit test to match with unique ID validation, have character limits for the field and have non updateable ID properties.

* + 1. Defend the overall quality of your JUnit tests. In other words, how do you know your JUnit tests were **effective** based on the coverage percentage?

I would say testing that Junit provided would be a good 80% coverage. I gave it that number because it does a good job on covering stuff like critical path and edge cases. Test like ‘testTaskServiceAddTask()’ gave it a good coverage at its core. I never want to give it 100% because there are always going to be something new that’s overlooked but I can say I can give it an 80%.

* 1. Describe your experience writing the JUnit tests.
     1. How did you ensure that your code was **technically sound**? Cite specific lines of code from your tests to illustrate.

To make sure its technically sound I usually follow industry standards and best practices. Based on that it had proper structure and validation I used ‘testAddContact()’ to check a name and a phone number.

* + 1. How did you ensure that your code was **efficient**? Cite specific lines of code from your tests to illustrate.

It was efficient because I was able to get rid of redundant code. For example I used ‘testTaskServiceDeleteTask()’ to check if it exist, after and before deletion. Which gets rid of making another code to check that.

1. **Reflection**
   1. Testing Techniques
      1. What were the **software testing techniques** that you employed in this project? Describe their characteristics using specific details.

For this project I used Black Box testing which focuses on input and output behavior and an example of that would be this line of code I used. ‘testAppointmentInvalidDate()’ which focuses on the correct output. Another one I used was Boundary Value testing which targets edge cases inputs and the code I used was ‘testTaskNameMaxLength()’. Lastly I used Negative testing which ensures the systems responds to the invalid code and the code I used was ‘testAddInvalidContactPhone()’

* + 1. What are the **other software testing techniques** that you did not use for this project? Describe their characteristics using specific details.

I didn’t use Whitebox testing and integration testing. Whitebox testing is focuses on internal code and I didn’t do that. For Integration testing its how different components interact and I didn’t do that either.

* + 1. For each of the techniques you discussed, explain the **practical uses and implications** for different software development projects and situations.

Black box testing is most use to make sure the system is following the requirements. White box testing is more used for debugging put more specifically its internal. Lastly its integration testing and that’s best used with different components working with each other.

* 1. Mindset
     1. Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ **caution**? Why was it important to appreciate the complexity and interrelationships of the code you were testing? Provide specific examples to illustrate your claims.

My mindset was baby steps, not to rush it that its not a race. The moment I thought I have to race or finish this fast it was all over. With that in mind I was always cautious and testing my code every time I finished a line. Just to make sure I catch it right away to get rid of any bugs that may have crossed my path. I also had coverage and understand that the code has to be complex in order to work smarter and not harder so I used this, ‘testUpdateTaskInvalidDescription(‘ to ensure I have coverage.

* + 1. Assess the ways you tried to limit **bias** in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code? Provide specific examples to illustrate your claims.

There is always going to be bias especially when its your own work but to limit that, its important to test your work as soon as possible. This will help eliminate as much bias as possible so I added this ‘testAddTaskWithDuplicateID()’ to test the add task intensely. That way it does its job and there isn’t any looks good to me I wrote it talk.

* + 1. Finally, evaluate the importance of being **disciplined** in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field? Provide specific examples to illustrate your claims.

Its important to be disciplined since this can give you an encouragement to cut corner and take the easy road. I promise you by doing that the quality of code would be low. It lead to even more time debugging and figuring out what you did wrong all because you decided to cut corner to “save time”. I used ‘testTaskServiceEdgeCases()’ to make sure that the low probability cases were handled efficiently.

Work Cited:

GeeksforGeeks. (2023, December 6). *Software testing techniques*. https://www.geeksforgeeks.org/software-testing-techniques/

DePuy, T. (2018, May 25). *How to align technical details with business requirements*. SD Times. https://sdtimes.com/softwaredev/how-to-align-technical-details-with-business-requirements/

Marget, A. (2024, August 6). *Development and test environments: Understanding the different types of environments*. Unitrends. https://www.unitrends.com/blog/development-test-environments