**Project Two**

The role of Unit testing is to reduce risk and help increase the quality of the code by finding defects in the software product. The unit testing approach I used for each of the three features was Junit testing which is considered to be a type of dynamic testing. I made sure that the software requirements and unit testing were completely aligned.

The testing process was organized, I created a test for each class in Contact, Contact Service, Task, Task Service, Appointment and Appointment Service. The testing process was organized to make sure I kept track of each class as I tested each class. The JUnit tests were effective based on the coverage percentage, because I created a test for each class in the java files.

The overall quality of my Junit tests has improved since module three when I first wrote the Contact test Junit test code with the assertions. I had never written Junit test code with assertions, so I had many errors. Once I had written the Junit tests without errors I added more Junit test code to both the contact test and contact service test java files. When I was working on Project One I corrected almost all the errors I had in the ContactTest and ContactServiceTest Junit java files.

JUnit testing was the software testing technique that I employed for milestones three, four and five. I was able to create tests incrementally that measure progress by using JUnit tests, so that I could see the results immediately. When I made any changes to the code of the Junit test I observed the result of that change to make sure the desired result was achieved.

The overall quality of my Junit tests are great, because my code coverage was 100% for five of my Junit test files; ContactTest, TaskTest, TaskServiceTest, AppointmentTest and Appointment ServiceTest Junit java files. Every part of the code in the test cases and each test case for each requirement was included. In the ContactServiceTest Junit java test file I added more assertions to the Junit test for newContactTest and newContactTest test ran with errors caused by the three lines of code listed below.

()->assertEquals("1235559999",service.getContactList().get(0).getPhoneNumber()),

()->assertEquals("1235559999",service.getContactList().get(1).getPhoneNumber()),

()->assertEquals("1235559999",service.getContactList().get(2).getPhoneNumber()),

I commented out these three lines, and the Junit test for newContactTest, in the ContactServiceTest Junit java test file ran without any errors or failures.

My Junit tests are effective because the Junit tests I created ran numerous times without any failures and errors after I completed writing the code for the ContactTest, TaskTest and TaskServiceTest, AppointmentTest and Appointment ServiceTest Junit java test files. I compared my code to the list of requirements to make sure that the code I wrote met all the requirements.

I took steps to ensure that my code was technically sound by using arrays, a type of data structure. In the ContactService class I used the array, private List<Contact> contactList = new ArrayList<>() to make a list for the strings. I also used the array private final List<Task> taskList = new ArrayList<>(), in the TaskService class and in AppointmentService I used, final private List<Appointment> appointmentList = new ArrayList<>(). I also properly named all the classes and methods. In the example below I used assertions to test the code to ensure the code met the software requirements.

Text

Description automatically generated

I ensured that the code was efficient by declaring the variables before

Initiating them. In the example below I first declared the variables contactId, firstNameTest, lastNameTest, phoneNumberTest,addressTest and then I initialized those variables.

Text

Description automatically generated

JUnit testing was the software testing technique that I employed for each of the three features. JUnit testing is used to apply unit testing in java to improve the quality of the code and accelerate programming speed. JUnit testing is a regression testing framework that checks that all problems have been successfully addressed when the environment has changed (Junit - Test Framework, n.d.).

The other software testing techniques that I did not use for the milestones were non-functional testing and system testing. Black-box testing is a type of non-functional testing which examines the usability of the software and performance under load and stress (Hambling, 2015, p. 53). System testing focuses on the behavior of the whole system/product as defined by the scope of a development project or program, in a representative live environment (System Testing, n.d.). System testing takes place after integration testing and unit testing.

Non-functional testing is a method of software testing that examines a software application's non-functional attributes. Non-functional testing can improve the software’s reliability, performance and accessibility (Nanda, 2021). System testing is performed to evaluate the complete system's compliance against specified requirements (System Testing, n.d.). In System testing, the functionalities of the system are tested from an end-to-end perspective. Tests are designed and performed to determine whether the product functions and fulfills the requirements that were set.

The mindset I adopted working on this project was to carefully compare my code to the list of requirements. My objective was to create a test for each class in Contact, Contact Service, Task, Task Service, Appointment and Appointment Service. I think I was very cautious, I added more Junit test code to the java files, once I had written the Junit tests without errors. I attempted to limit bias in my review of the code by running the JUnit tests numerous times without any failures and errors after I completed writing the code.

Being disciplined in my commitment to quality is essential to creating a software product with less defects and errors. It is important not to cut corners when it comes to writing or testing code because of the impact that defects and errors can have on the software product. Customer’s using the software product will lose confidence in the company making the product and may choose to no longer purchase or use the product resulting in financial losses for the company.

**References:**

1. Hambling, Brian Morgan, Peter Samaroo, Angelina Thompson, Geoff Williams, Peter. (2015). Software Testing - An ISTQB-BCS Certified Tester Foundation Guide (3rd Edition) - 1.7 Fundamental Test Process. BCS The Chartered Institute for IT. Retrieved from https://app.knovel.com/hotlink/pdf/id:kt00UC2H22/software-testing-an-istqb/fundamenta-fundamental
2. Junit - Test Framework. (n.d.). Retrieved April 3, 2022, from https://www.tutorialspoint.com/junit/junit\_test\_framework.htm
3. Nanda, V. (2021, May 13). What is non-functional testing? . What is non-functional testing? Types with example. Retrieved April 3, 2022, from https://www.tutorialspoint.com/what-is-non-functional-testing-types-with-example
4. System Testing. (n.d.). Retrieved April 3, 2022, from https://www.tutorialspoint.com/software\_testing\_dictionary/system\_testing.htm