### Southern New Hampshire University

# 7-2 Project Two: Summary and Reflection

### Bairon Gomez

### Professor Lewis

### CS 320 – Software Test Automation

February 18th, 2023

The software requirements aligned with my approach by focusing and understanding what the customer desired to establish as requirements. Two requirements that were seen to be implemented repeatedly are the length of the contact ID having to be less than 10 characters and not having the value of null. These requirements were checked by implanting the following code: if (contactID == null || contactID.length() > 10) {throw new IllegalArgumentException ("Invalid Contact ID");. The length requirement is checked by contactId.length() > 10 and the null requirement is checked by ContactId == null. These methods were tested by creating two different classes for each requirement that were represented by contIdExceedsLength and contIdNull. Each contact ID input would be checked to validate whether the costumer requirements passed or failed. The same process is applied to other requirements imposed by the customer for the software.

Other than having the importance of the software requirements guide my work, I also ensured that the quality of Junit tests were efficient and clean. The quality of the Junit is reflected by the percentage performance which was 100% in eclipse when the Junit was implemented. There were different Junit tests created to test the different requirements implemented by the customer and all of the tests were passed with 100% coverage. The coverage percentage represents how the code performed. Since the coverage is at 100%, this provided me with positive feedback that the customers requirements were met. In addition to the coverage percentage, I also ensured that the variables used and created would be understood by a reader who is assessing the code. This will allow reader and programmer to focus on the functionality of the code rather than to waste time on determining what each variable represents.

Testing techniques are important to understand to ensure that the techniques applied are efficient and effective within the code. The test that was incorporated in the requirements are known as white box and unit testing. White box focuses on the internal operations of the program. Before focusing on how to test different inputs within the code the user must understand what should be expected as the output. Once the output is known for each input it simplifies the process by enabling the programmer to focus on the inputs and the reaction is should trigger based on whether the inputs are valid/invalid within each test. It is important to have valid and invalid inputs to validate whether the internal process of the test is working according to the requirements. Unit testing on the other hand focuses on examining sections individually to ensure that they are implemented properly. Unit testing was applied by entering invalid inputs intentionally to observe the reaction. If invalid inputs resulted in a exception exerted it meant that the test was working appropriately.

Although I focused on two software techniques, there are many other techniques that were not used due to relevancy of the techniques in my work. Some of the techniques that were not implemented are known as security testing and usability testing. The emphasize on security testing is to provide safe and secure data into the system that is being used as an input. The usability test focuses on ensuring that the users of the system will have the ability to navigate through the system with ease. Both tests can be applicable to the software that is being developed for the client, but in this case it is not the primary focus. Security testing can be used in the program, but it is something that would have to incorporated separately. The customers personal information will be required in the future which would require the data that is being used as an input to be secure to prevent corrupted data from corrupting the system. On the other hand unit testing can also be incorporated once the application is completed. The usability of the application can be determined once users have access to the final application. Feedback can be provided about what the users find simple to access/use and difficult to access.

Furthermore, the white box tests will provide different assertions that the requirements imposed by the customer and system are met. If the requirements are not met the user will be given an error message that will inform them about the requirements that must be followed and where their mistake was found. White box testing can be found be implemented when creating restrictions over the length of a password, the characters that can be used in a password and so on. Unit testing on the other hand is imposed to validate the functions in the software individually.

Having and adopting the proper mindset while working on a project is important to understand what and how to develop code that will meet the requirements of the client. Each task in a project is different, but in this case it was important to be vigilant to establish the proper requirements established by the client. The mindset of a tester is what I adopted while ensuring that the tests created aligned with the customer's requirements. It was important to understand what was expected from each requirement imposed to develop a proper requirement. This practice enabled me to be cautious when developing the requirements. The practice of incorporating caution to develop the requirements into code reflected on the coverage report. If the coverage report returned 100% it meant that the code was effective and did not have any bugs within it.

As a developer it is important to know the quality of your work, but it is also important not to be biased about your code. This can lead to the work produced to contain overseen requirements that can lead to an incomplete project for the client. This can be prevented by assessing the code developed while and after it is developed from the perspective of a developer and tester. Looking at the work developed as a tester allows the developer to focus on what the outcome should be based on the input. This will enable the code to be analyzed throughout the development process to ensure that the work created is complete and meets the requirements. Overlooking the projector as a tester also enables the code to be assessed for flaws that are not so obvious and difficult to detect. Looking at my work forced me analyze my work and question whether I took the most efficient and clean method of coding. Some minor improvements were made after looking at my work through the eyes of a tester, but overall it help pinpoint and review the quality of my work. Although assessing the work I created helped me catch some bugs, It can also prevent me seeing and fixing code that I am not trained to find. This can lead to overconfidence that will prevent the code from being improved. One example can be looking into two different tests. Knowing that one particular requirement passed can lead to assuming that a different function with the same requirement will pass, but it is not assertive because the tests are runed separately.

Lastly, the importance of being disciplined to provide quality as a software engineer professional is highly important and will be reflected on the work of the developer. Taking short cuts throughout the development process of tests or functions will be seen in the end product. Short cuts and inefficient work will be found sooner or later by looking at the performance of the end product. All of the outcomes from cutting corners lead to jeopardizing the end product. Some outcomes are more devastating than other. The minor results can simply be that the system crashes or a particular feature fails. Depending on how difficult the failure it, it will cost the company an immense amount of money to repair and improve the current system to prevent crashes. Major damage can be made if the information from the users and company is leaked because the testing was done improperly. This will lead to the lives of many users becoming vulnerable and facing financial difficulties. An example of this can be seen when looking into a bank system. If the developer neglects to be consistent and maintain quality work it can lead to the bank system failing. Millions of users can be impacted by not being able to access their funds in the bank. This will prevent users from making payments such as mortgage payments, rent, medical, and so on that can be devastating for the users and the bank. The bank will lose their reputations and be liable for the damages done to their users. All of this can be prevented by taking the proper approach while testing code. Incorporating the agile approach and using resources such as other developers can save time and money for everyone. The agile approach enables the features of the system to be tested consistently to ensure that the end product does not contain flaws of bugs. Having a peer review your work can also help by having a different perspective on things that the developer may have overlooked.