CS 320 Software Test Automation & QA

Vince Taliaferro

Southern New Hampshire University

6/18/2023

The unit testing approach with specific software requirements is very important for building solutions. Reducing the risk of running into errors in the code and improving the quality while finding defects and getting them fixed. One example is Contact Service that had specified requirements that had to be met with a unique contact ID string that could not be over 10 characters, could not be null or updateable. Requiring firstName and lastName at 10 characters max and shall not be null. Also included was a phone field that is a string at 10 digits max. In the appointment service class, the description could not exceed fifty characters, and the description could not be null. The tests were used to give exceptions if the stipulations were not met. The coverage of the tests would confirm all those parameters were tested. The testing for the classes would indicate a high percentage of coverage. Contact service was tested at 88.1% while task service was tested with a percentage of 94.3% and appointment service was running at 91.3%

Using the correct amount of code needed to meet the requirements. Tests for adding, removing, updated, and null values were all tested through JUnit tests. We had to make sure that the code was up to par with the best practices and the correct naming principles. Testing for true and false statements would help to make sure the code was efficient and made correctly. The example below shows these assertions that were tested.



In the milestones contact, task, and account services. We used JUnit testing as our primary measure of efficiency and correctness. Testing each method for validation and for proper procedure. This testing allowed individual units to be independently tested with assertion in the comparisons of expected values and what the outcome is. We wrote the conditions within the code along with the assertions to were used to make sure that the conditions were met. Coverage testing for our system was the tool used for the rate of the code.

Black box test procedures are a strategy that could have also been used. This testing is for going into testing without previous knowledge of the system itself. The testers will analyze the software to identify defects according to the requirements. Integration is testing that brings together parts of the system to make an output.

The use of JUnit testing will include the ability to make clean code that will be assured through testing done during development. JUnit testing will be a tool that improves the quality of the system we are aiming for. Test driven development, JUnit testing would be beneficial to agile test units. Different methodologies can benefit from other types of testing and analysis of the best approach should be made. Finding the best method of testing should be met with exploration along with trial and error.

I did my best to get rid of bias while looking at the code even when the percentages of the JUnit tests were running high. I went through the tests and reviewed the code many times to see if the coverage could be improved. Trying multiple approaches and methodologies until the code seemed as efficient as possible. Naming conventions and principles were used to create clean and efficient code. Keeping it readable and usable for others involved in the development of the system. This kept my mind from getting too focused on one way of getting to the end goal.

Trying to remain focused on quality was a main concern for me in developing this challenging project. Overall, I enjoyed the process and difficulty that went into bringing all our code together for a final project. Trying to make the best product that I am capable of I focused on efficiency and readability. Having good fundamentals and exploring new ideas felt like the best approach to this project. I enjoyed learning about this testing method and gained a lot of insight into what it takes to get a project like this and hope to gain more knowledge into how to test different configurations of code.