Michael Ladderbush

Module 4 Journal

My testing approach aligned with the software requirements by steadily creating tests for each new function I created. When the requirements called for the contact and task services to be objects with specific variables I created those objects first in a class, then I immediately went and created test classes for those objects. The objects themselves needed default constructors that would take the parameters mentioned in the requirements and once I had finished those, I made tests for the getters and setters for each of those parameters, along with a set up test that would establish an initialized constructor with the necessary variables. Throughout the code each line that I would write I would then create a test case in order to prove that the functionality is properly working and makes sense.

I think the code coverage of my JUnit tests makes a strong case for the quality of the tests in general. As I mentioned before every line in the code has an equivalent test function in order to sequentially check that there are no problems in the generation of either tasks or contacts. Along with testing code coverage I also ensured that my code was sound by using industry best practices and maintaining robust objects and variables. The functions I used to create and update variables in both of the classes have specific checks that will throw and exception is either the length or presence of the variable data is invalid. Ensuring the code efficiency was also a matter of using best practices but also by keeping the code simple and modular. Most of the variables had very similar getters and setters along with checks to guarantee validity. The speed of the code is not reliant on any complicated algorithms and doesn’t have any unnecessary bottlenecks at any point. The code and tests both take milliseconds to compile and run with no slowdowns due to code errors. The quality, efficiency and robustness of the code can be seen throughout and is proven by the lack of errors, consistency and speed of execution.