Enhancement One: Software Design and Engineering

Ryne Williams

Department of Computer Science, Southern New Hampshire University

CS 499: Computer Science Capstone

Brooke Goggin

11/11/2023

The artifact that I chose for enhancement one was created in CS 320: Software Testing, Automation, and Quality Assurance. This code is a Junit testing code in Java that tests the functionality of six different files. The Appointment Service, Contact Service, and Task Service take user input to display, add, delete, or modify appointments, contacts, or tasks. Each of the Service classes have an accompanying class that holds error handling for the user input and the getters and setters for the objects in each array list. The original code had over 90% coverage for tests against the main code.

I chose this artifact because it was a good opportunity to put into practice coding concepts that I learned from previous courses, as well as an opportunity to challenge myself with a concept that I learned recently and put it into practice. I believe that this enhancement showcases my ability to understand the syntactical and structural differences between two different coding languages and how to convert one to another. Converting the code from Java to C++ enhanced this code to improve its portability, allowing it to be implemented in more applications.

I feel as if I met the course objectives that I had intended to meet with this enhancement. I was able to successfully convert the code from Java to C++ and ensured that all test cases successfully passed. I feel that this enhancement accurately employs strategies for building collaborative environments that enable diverse audiences to support organizational decision making field of computer science. I also believe that this enhancement successfully develops a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

When creating this enhancement, I ran into some issues with the “#include” tag for linking the different files. After researching the issue and trying multiple solutions, I ended up realizing that I needed to move the class declarations into a “.h” file and implement them in the “.cpp” file in order to ensure that the compiler did not have issues with class redefinitions due to “#include” tags in multiple files. This was my largest challenge in this enhancement and took the longest time to resolve. Other challenges were trying to find out why I was having issues with some of the tests failing, which turned out to be simply mismatching error messages or date formats.