Brian Engel

Module Five Journal

In each of the milestones, the main type of testing I did was functional testing, using unit testing in J unit. I would create test cases from the requirements and write a specific unit test for each test case. This ensured that the output of each function matched what the requirement was. I suppose I also used a very small amount of integration testing as well, since some of the functions that I wrote call other functions in them. This occurs most frequently in the way I designed the constructors. They set the id since it has to be unique, but then call a separate function to set everything else, such as name, date, and description. Also, integration testing was also used in between classes. An example would be that the TaskService class integrates the Task class into it by having an array list of Task objects. If the Task class didn’t work, neither would the TaskService class. One other type of testing I did would be regression testing. Every time I made a change to anything, I would run the entire test suite again to make sure everything was still working.

There are a bunch of types of testing I didn’t use as well. I didn’t do any sort of performance, stress, or load testing. To do this you would have to test the application with large amounts of data that you would encounter in use. This would show if the system were stable, ran fast enough, and how much data it would take to break it. There was also no usability testing, since there is no user interface to interact with. There was no acceptance testing, since no stakeholders passed off on it, and no Alpha or Beta testing since the application isn’t finished and definitely not ready to be tested by users.

The practical use of these techniques really just depends on where you are in the project. Functional testing and unit testing are going to be most useful when you are first writing code and making sure it passes the use cases. Integration testing is for when you start putting classes and packages together to make sure they work with each other. Regression testing is most useful when you change code and have to make sure you didn’t break anything while doing it. All of the performance testing would happen after you have a working application. Usability testing could be included with alpha and beta testing, and honestly anyone working with the user interface could give feedback that would count as usability testing. In alpha testing, the goal is to find and fix as many bugs as possible before it gets passed on to a group of users and outside testers in beta testing. This is real world conditions but on a limited scale and done to discover any issues with different systems and environments. After this, if everything works correctly, there is only acceptance testing, which is done by the stakeholders to make sure the software meets all of their requirements.