Elliot Putnam

CS320 – Software Test Automation & QA

Module Five Journal

Across modules three through five, I have worked on creating small classes in Java that allowed me to test my introductory skills for testing, specifically using Junit tests. For my three milestones, a few testing techniques that were implemented would be; Boundary Testing, because some of tests were designed to check if certain parameters were in bounds, such as a character limit for an ID which could not be null, nor over 12 units (1-12). Unit testing, since I was using a predetermined data set within my functions, rather than randomized input. Although minimal, I consider these milestones to contain acceptance testing, because they were designed with determined requirements and were written to provide a result.

Some techniques that I did not use during my milestones would be regression testing, where the system is tested after changes are made. An argument could be made that I may be testing for regression, however I believe it’s more along the line of debugging with the ‘changes’ I was making. Another test would be performance testing, because I no way was I monitoring system speeds or stability, nor was I developing or testing to see if my program could properly scale.

Boundary Tests are important when dealing with any program that works with number caps. A good example would be in a video game when the character you play can only reach a certain level or hold a certain amount of gold coins. I have played games where certain amounts of experience or gold added at once would break the game and push a ‘capped’ number into a null error. This is because the system designed to intake and output the data had flaws.

Acceptance tests are needed during all software development projects because without them, you may be delivering a product that will be rejected or create a sour business exchange. Testing the program to match the users’ requirements ensures that the customer received what they expected.

Performance testing can be beneficial to developers looking to deliver a product that parses real time data. Server based applications and peer-to-peer software often requires near instant transaction speeds (for the customer). A function that runs too many times, or gets stuck in a loop, or doesn’t work at all, can hinder cycle times and cause back ups. That is why performance testing is important to ensure that the program is running optimally from the code, to the hardware, and through the web.