Tyler Morgan

CS-320 Software Test Automation& QA

Module 5-2 Journal: Software testing Techniques

* What were the software testing techniques that you employed for each of the milestones? Describe their characteristics using specific details.

For Each of these milestones we had to employed JUNIT testing using JUNIT 5 libraries. This is a form of white box testing, which uses full transparency, allowing the tester to see and test the inner workings of the code. JUNIT 5 is a test plug-in that you can run using different applications such as Maven, Gradle, or through the Eclipse IDE itself. I used the IDE by importing the JUNIT 5 library and running the tests that way. JUNIT 5 encompasses the features of previous versions by using the JUNIT Vintage sub-project. It also uses 2 other sub-projects which are JUNIT Platform, and JUNIT Jupiter. These sub-projects define the TestEngine API for developing new testing frameworks, and hold all new JUNIT annotations TestEngine implementation, respectively. We also ran coverage tests to see the percentage of our code that was covered and tested by our JUNIT tests. Additionally, we always use static testing while working with software development. We don’t have to execute code to preform static testing, instead, we can just read through and fix any noticeable grammatical or syntax errors.

* What are the other software testing techniques that you did not use for the milestones? Describe their characteristics using specific details.

Another form of testing that we didn’t get into is black box testing. This refers to the type of testing that does not allow transparency into the inner workings of the code, and only allows the tester to test the user-end of the software. This is used in most modern-day software applications to test input values versus output values. This type of testing is used throughout all stages of the SDLC (The QA Lead & Boog, 2019)

* For each of the techniques you discussed, explain the practical uses and implications for different software development projects and situations.

When looking at white box testing vs. black box testing, they both have their practicalities. Black box testing is very important for validating user input and ensuring the correct output, meanwhile, white box testing ensure that the functions built from the black box testing work. Black box testing is a building block of software testing. With a goal of quality assurance for the end user. The goal for white box testing is to assess all possible cases and scenarios of the code’s functionality. We use this on smaller targets in the code such as functions and methods to ensure flawless functionality (Ashtari, 2022). We explored very few types of software testing during these milestones, but there are many very important types of testing being used today.

**References:**

Ashtari, H. (2022, September 29). *Black box vs. white box testing: Understanding 3 key differences*. Black Box Testing vs. White Box Testing. Retrieved November 27, 2022, from <https://www.spiceworks.com/tech/devops/articles/black-box-vs-white-box-testing/#:~:text=Black%20box%20testing%20is%20a,and%20analyzes%20it%20during%20testing>.

The QA Lead, & Boog, J. (2019, September 5). *9 types of software testing in software engineering*. The QA Lead. Retrieved November 27, 2022, from <https://theqalead.com/test-management/types-of-software-testing/>