**Software Testing Techniques**

Matthew A Keaton

Southern New Hampshire University

CS 320: Software Test, Automation QA

Professor Federico Bermudez

April 6, 2025

* **What were the software testing techniques that you employed for each of the milestones? Describe their characteristics using specific details.**
  + For each of the milestones I employed the software testing technique of unit testing, specifically using Junit testing in Java. Junit testing is a very popular framework used in Java that is open sourced and used to write and run tests (Jakubiak, 2022). Unit testing allows me to test individual pieces of the code such as methods or classes to verify their functionality. Each test targets a specific function, for example, the appointment ID should not be longer than 10 characters and should not be null. Junit testing will check to ensure that the method created for setting an appointment ID will meet those requirements and throw an error if it does not. These tests are automated and quick to run, which helps ensure efficient validation during development.
* **What are the other software testing techniques that you did not use for the milestones? Describe their characteristics using specific details.**
  + A software testing technique that was not used in the milestones is the different testing annotations. For example, a specific annotation that was not used is the “@After” annotation. This annotation identifies a method that should run after each test method in the class (Jakubiak, 2022). These annotations are how the testing framework identifies and handles certain methods or classes during testing. Another testing annotation that was not implemented into the milestones was the “@Ignore” annotation that specifies which method should not be ran while testing.
* **For each of the techniques you discussed, explain the practical uses and implications for different software development projects and situations.**
  + The testing techniques mentioned above are very useful in different types of projects. Unit testing is great to use for checking individual pieces of code to ensure they function as they are expected. This is especially useful in projects that are updated often since the tests that are run are quick and simple. Using the “@After” testing annotation is useful to implement in projects that needs to close files after being opened or resetting data to start the next test with fresh data. The “@Ignore” testing annotation can be very useful to use in big projects where the code is not ready to be tested or is incomplete. This allows you to still run tests on other parts of your code that are ready to be tested without having to remove or finish the parts that aren’t ready. These testing techniques help make it easier to manage each test and ensure everything is running smoothly.

**Reference**

Jakubiak, N. (2022, December 6). *JUnit tutorial: Setting up, writing, and running Java unit tests*. Parasoft. <https://www.parasoft.com/blog/junit-tutorial-setting-up-writing-and-running> java-unit-tests/