Journal 5

CS-320 SNHU

06/04/2023

Tammy Hartline

* **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 have completed thus far, I have added, or tried to add further testing methods to each suite. For the most recent milestone completed, I relied heavily on unit testing to test the functionality of the individual code components using JUnit. The primary characteristics of unit testing is to isolate application components, through providing input, and then checking the output is as expected. To accomplish this, I relied heavily on the assert and assertions libraries. I also utilized acceptance testing. I accomplished this by making certain I had a clear understanding of the requirements/restrictions for this project and designed my testing suite and program files around the instructions given to ensure that they meet/adhere to “customer” standards.

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

A few of the techniques I have yet to use for the milestones include regression testing, performance testing, integration testing, and security testing. While I believe each of the ones I listed is a crucial part of software testing, I am still quite inexperienced at how to actually implement these tests in a real-world scenario, aside from regression testing, (light) through use of observing the runtime difference each time a change is made to the program. Regression testing can be used to verify whether newly added or updated functionalities impact the program in a negative, positive, or neutral way. Performance testing is similar, but more focused on the speed, stability, and responsiveness of the system. The main purpose of performance testing appears to be the ability to better understand and highlight areas of program code that can be altered to provide better performance. Security testing is used to ensure that a system remains healthy, through limiting access to approved users only, protection from hacking, and other security threats that technology is vulnerable to. Lastly, integration testing is the one that I am most eager to learn more about. Its primary characteristics include verifying the interaction between modules, validating data flow through the interface, and establishing any defect areas in the system as a whole, by checking that the individual components that are meant to integrate, output correctly and operate as desired. In the most recent milestone completed, I tried to find a way to test that the Appointment.java file would work with the AppointmentService.java file. But I did struggle to get my tests to execute at the same time. Instead, I would run the AppointmentTest file, and it would return coverage for that file and the Appointment file but showed the AppointmentService and AppointmentServiceTest files coverages to be 0% and vice versa. My thought process for this is that a greater understanding of integration testing would have allowed me to execute one of the tests, while the other test simultaneously executed as well, but I may be mistaken and looking at it from the wrong point of view.

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

I believe the techniques each hold a practical and real-world use that would depend on the scenario and what it is one is trying to accomplish with the program as a whole, but also with the testing. I find unit testing to be the easiest to implement at the moment, as it makes the most sense for the given projects in this course so far. I stated earlier about my reason for being more interested in integration testing. Security testing is one that I believe every computer science, data science, or technology related field major should learn, and learn well. They should be able to apply these practices in order to keep data confidential and only distributed on a need-to-know basis. If more people in the technology field understood security testing and cyber security as a whole, I believe we would have far fewer systems so easily hackable. System, Security, Performance, Regression, Acceptance, Integration and Unit testing, (Of which I am sure I forgot other testing methods), each have a key role in development. Which of these and how many of them we use will often depend on many outside variables, such as the specific project needs and requirements, release schedule, budgets, software features and the software’s intended use. I believe the key concepts we are learning in this course will help prepare us for use of these testing methods that will help us prevent program defects, while also reducing system risk, improving code quality, and enhancing the overall user experience when the end product is released.