Zachary Huff

Southern New Hampshire University

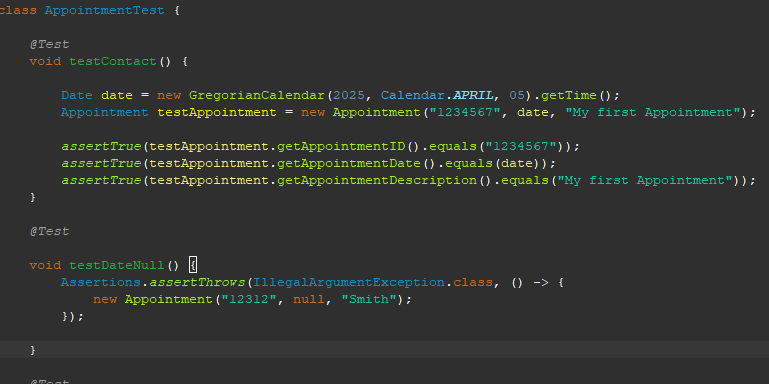
CS320 Software Test Automation

Dr. Errol Waithe

January 23, 2023

Summary and Reflections Report

My unit testing was based completely around the functional requirements given to us for each assignment section or project. If we had variables that could not be longer or shorter than a certain length for example; I would write a test for that. That is how I went about writing my JUnit classes, simply based around the requirements.



This image is taken from my AppointmentTest class. The first JUnit test accounts for one of the functional tests given to us on the last part of Project 1. It checked to make sure that the date the user would not input a null pointer for the date value, a similar test is written further down to test if the user’s appointment date is valid (not in the past). This is how each test throughout the course was written. They were quick and efficient to write, and they tested what was required of the software, nothing more and nothing less. My JUnit test improved over the course though, in the beginning two projects I was using more of a hardcoded based approach when it came to the non-testing classes. Because of the feedback I was given I went back and fixed my other two parts of the project and made sure part three was not hard coded. After doing this I had to change up some of the test cases I had written about as they no longer functioned properly, this ended up taking the most time.

Other than JUnit tests there were a couple instances where I created a main () driver class to test whether my functions/methods were working how they should be. This allowed me to quickly test whether something was going to work or not before I wrote a test case for it. For this type of smaller project this seemed to work more quickly for me. This approach was not necessarily needed until I started to make the classes more user friendly and less hard coded. For this assignment I did not do any testing above what was necessary. In my last class we just maven to perform security testing which could have also been done with this project. Also, I did not do any performance testing because the project is so small speed really is not a factor. For larger projects these testing practices would be especially important. Security and speed are among the most important types of testing after making sure the project does what it is supposed to do. When projects become exponentially larger and larger, they introduce more security vulnerabilities and speed becomes a huge factor.

It is important to employ caution when testing code because like the question states: “The problems can be complex and have many interrelationships.” This partially answers the question. For example, when testing a certain part of the project it may run fine in a set amount of unit tests but when implemented into the entire project may fail to check for errors correctly. This may give the developers a false sense of security when testing and lead them to believe certain code is not the problem when it still could potentially be causing the errors. Because we are all human, bias can occur in development just as it can anywhere else. You may think your code does not even need to be tested because you are one hundred percent sure there are no errors. I am guilty of this just in my short programming experience. There are projects I have turned in without doing any major testing and was docked points because of the errors the professor had found. I could not believe those errors existed but sure enough when I opened it back up I saw where I had made the mistakes. This would have easily been noticed by a second pair of eyes, but unlike drafting a paper, most people do not know what they are looking at when it comes to coding let alone know how to find errors.

Being disciplined in your commitment to writing quality code can be difficult at times. I personally experienced this a couple of times throughout this course. There were times when I had little time in the week to get my assignments done. I wanted to just blast through the assignment and do the bare minimum to get a passing grade. When this happened, I just decided turning it in late would be better than turning in an assignment that was half-baked. It is important to dedicate more time than you think when writing code, cause often you will run into unforeseeable problems. These problems can end up taking many hours to solve if you do not know exactly what you are doing, which is an issue, I would assume, most students at university have. Ignoring these problems and cutting corners to reach a deadline could potentially result in much larger issues down the road. An effective way to avoid this would be to allocate more time than you think you are going to need to these projects. This way you will not be disappointed when you take longer than expected and when you finish before you will feel satisfied. This is easier said than done though, because in the real world it is hard to set deadlines. If this happens to be the issue, then it would be wise to ask a colleague for help or let management know you will need more time to finish. Unfortunately, this is the best advice I have about where I stand. I have never held a real job in software development, so I am sure it is much more complicated than this.

Resources:

*SDLC - Overview*. Tutorials Point. (n.d.). Retrieved February 26, 2023, from <https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>

*Software testing methodologies*. smartbear.com. (n.d.). Retrieved February 26, 2023, from <https://smartbear.com/learn/automated-testing/software-testing-methodologies/>