Daniel Anderson

Software Test Automation

6/16/2023

**Project Summary**

The client requested that I develop an application that would cover different parts of software to help them with their own clients. Within the application I had to create different classes for Contact, Task, and Appointment. When I had to test the requirements, I set up a test section for each class to test that the standards for each one was properly set and were followed as programmed. The main purpose of the testing was to ensure they could enter the information correctly with in different parameters and if the parameters were not meet it would kick out an error message notifying the user of the error. Another type of type of testing I did was to visual check the code to ensure that it was written properly and there were no errors that could cause bugs later down the line. Junit testing was a very effective tool in verifying that my code was effective and efficient. I was able to get my Junit testing to check all the requirement so the software and to ensure that everything would be able to be entered correctly and would kick back an error message if something was entered incorrectly.

Using Junit was a very interesting experience for me, and it was able to teach me a lot of different ways to test the coding I have typed. One area that I feel my code was done well was under the appointment classes. For the appointment test class, I went through to ensure that every possible way something could be entered was tested. When looking at the appointment service test class I did a good job at testing if an appointment could be made or if one was already created under certain conditions. Much like my technically sound lines my efficient lines are nearly identical. Overall, after looking at my classes I feel that my test classes are written more effectively and accurately than the classes themselves. This has many positives but also a lot of negatives at the same time. My code shows I can test everything properly, but I did struggle to create the base classes efficiently which can the test classes some issues. One of the software testing techniques I employed during these assignments was just Junit testing. Junit testing is great for testing the components of the classes to ensure that they are working correctly and properly as intended. Another type of software testing I did was manual testing; manual testing is not as effective and Junit testing, but it allowed me to review my code to ensure its functionality. After reviewing the project, I can see how other types of software testing were utilized but were more merged into the other types of testing I did. Performance testing was a merged type of testing that showed that the classes would run properly, and they would run as intended. The last type of testing I feel was used but was merged in with manual testing was boundary value testing. This type of testing helps identify the inputs of your classes to check if they are working properly, which was a huge portion of this assignment. All types of software testing can be used for almost any project you are doing; however, you should really narrow done would. You need the testing to achieve based on what you are coding to get the most out of each test. A great example of this would be this project does not require security testing as we did not code anything to do with security into this assignment. This does not change the fact that security testing might not be used later when the project gets pushed to its next step there was however no reason to use that type of testing for the sections, we were programming.

While working on this project I had to take a certain mindset to ensure that I was coding the classes correctly. I did have to approach many things with caution to ensure that the classes would communicate with the correct test and that they would be tested correctly. This approach to must take each class as a separate entity before merging them to a working test format. How each class had its own test class shows the importance of how the classes work together to create a working code. The best way to show this is by looking at each class such as Contact, Task, or Appointment and seeing that they have a matching test class to ensure they are each working properly. When reviewing my code, I thought I was doing a great job with the base classes and upon looking at them on the first pass I had seen little to no errors in how my code was being written. This is where my bias meets the reality of how my coding was going, my code was doing great in some areas, but I was struggling in others. My Contact class is a perfect example of how I thought I was doing very well in my coding but was missing a lot of key components that not only would help the class run better but would also meet the client’s needs more effectively. Having bias in your code and thinking there is nothing wrong with it can cause many issues down the road because you can overlook certain errors in your coding that can hinder the effectiveness of your code. Ensuring your code is written correctly should always be a main concern of any software programmer. Anytime you make any code or even adjust any code you should always take the time to ensure that it is done correctly and neatly to ensure its usability. This is why it is important to not skip any testing or reviewing that can be done to your code because checking your code even after it is complete should always be a main priority throughout any project. I feel like the best way to avoid any issues down the road and to ensure my sanity as a programmer is to not only deploy different types of testing but to have someone else review my code as much as possible. I feel like by having a second set of eyes will really help keep any bias in check and ensure that the code is done to the best of my abilities and free from as many errors as possible. Throughout this class I learned that using multiple testing procedures is important and should be implemented as often as possible to avoid any and all issues that might occur.