Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 04/26/2018

Q4-d. What are some testing techniques we learned yesterday that can simplify the amount of tests that you would need to write (2 paragraphs.)
Ans:

The first testing technique is the equivalence class testing. This technique can separate the input domain into several sub-domains. In each sub-domain, the assumption is that the logic and the functionality are the same. Consequently, testers do not have to execute the test with all the possibilities within one single sub-domain. Due to this technique, testers only need to perform at least one test in each equivalence domain, and this can significantly reduce the number of tests.

In addition, boundary value testing is another technique to simplify tests. In the white-box testing, testers know what the boundary values are through observing the source codes. Consequently, they can only generate the test cases near the boundary values because values near the boundary are liable to error. For example, a grading software only allows users to input scores between 0 and 100. Testers do not need to perform tests with every single value from 0 to 100, and they only generate the test cases with -1, 0, 1, 99, 100, and 101. This can efficiently reduce the number of tests.

Q4-e. Is it possible to have 100% code coverage? Is code coverage a good testing strategy? (1 paragraph)
Ans: A 100% code coverage is possible. Developers should be able to create the tests that go through all their codes. In my opinion, code coverage is a good testing strategy because every single line of codes is executed. If the typos or logic mistakes exist, they can be immediately indicated. However, even though the test with 100% code coverage is performed, and no bug is found, this does not mean this code is absolutely bug-free. If the code does not match the requirement, it is still an issue.

Q4-f. If you just got hired to be a tester at Microsoft in their SQL Server division, how would you do your job? (1 paragraph).
Ans: At the beginning, I would try to understand the SQL Server as much as possible, such as the mission statement, the mission objectives, the tables, the fields, the relationships, and the business rules. The more I understand, the more efficient test cases I can generate. After that, I would start to create SQL Server unit tests and perform the tests. At the same time, I would do the version control for my test cases and test projects. This would help me

Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 04/26/2018
organize my work. Since the test platform will be upgraded sometimes, I would also update according to the changes of the test platform. If I had spare time, I would extend features of my test tools and learn more about troubleshooting skills related to my current testing.