CS Journal - Tester

CS 250 – Software Development Lifecycle

By: Brandon Rickman

Southern New Hampshire University

**February 3, 2019**

A tester in the Agile Development lifecycle is required to ensure programming functionality is correct, methods and process run as intended, and the client’s expectations match with the programs output. An Agile Tester has many functions and roles during the entire project but helps to hone the team’s skills during each sprint by trying to break the developers code. A good tester coupled with a good development team will lead to quick idea corrections or adaptations, an overall better performing team, and a team that is completely versatile.

The best user story elements for a Tester to have is the true picture of what the result should be. A Tester doesn’t need to be concerned as much with the final look of the project, however, this does need to be checked. The Tester needs to be concerned about the final functionality of the project or user story. A good question for the user to ask is, “What should this story do for the user?” Not only does the Tester need to be skilled in programming, but also in data manipulation, writing out test cases in English terms for anyone to read. The Tester needs to be ready to question the Product Owner on specific details from the client’s point of view.

During this exercise I found that the user stories presented were missing how the story was to function. I created them as a webpage and I believe this is what the result is, but I could not be sure due to the lack of detail in the email or in having a previous worked user story. If we could’ve gone back to the user stories we created and expanded them into the tests it would have given us more information to work with and created this entire story line. With only the email to go off of and without a product backlog developed for this story line it created some confusion as to what was expected. Ultimately, I was able to understand the purpose for this lesson and develop tests to conceptualize this lesson.

As a tester I would gather more information through meetings or follow up emails with the Product Owner and the SCRUM Master. These two resources would help to ensure I had the client’s vision and purpose in mind. I would also follow up with the Developer(s) to ensure that the result I was getting were the same as they were expecting when they wrote the code. Sometimes when writing code, you tend to get lost in the repetition and forget that each story has is own purpose or vision. A product backlog of the entire project would be a great resource to have to clear up any of the confusion, or the ability to use the product backlog generated in module 3. I know that uniformity is the best answer to writing a lesson plan and teaching several different students, but the direction seemed a little askew of what was required as the user story. By going on the “Tester’s Email” only there was not enough information to complete a test without developing some of the story’s functionality ad lib.

It was refreshing to complete this task though with the template. The template made it easy to understand all the parts necessary to breakdown any user story and complete and conduct a test. I did like that the test were first written out in English form, and not just code tests. Although there are times that coded Unit Tests will need to be conducted on projects, with the written English form it helps to create a set of rules for the Unit Test and creates a way that others can quickly understand what the test does and if the user story did what it was supposed to do.