Final Project: Retrospective

Thomas Buettell

CS 250 Software Development Lifecycle

June 15, 2022

This course saw me be able to take on the roles of all members of the Scrum Team. This includes the Product Owner, Scrum Master, Tester, and Developer. In doing so, I was able to learn how important each role is and important in understanding what each member of the team brings to the table. Using the Agile method, these team members must be open to communication and proficient in communicating to the other members of the team to facilitate proper team cohesion. Of all the roles listed above, each member of the team had a specific area that they specialize in to make the team work effectively.

**Product Owner**

The Product Owner handles most of the high-level decisions of the project and give direction to the team based on their communication with the client. The Product Owner stands as an intermediary between the rest of the team and the client, so communication at this level is mandatory and incredibly important. While in this role, I was responsible for utilizing my team effectively and clearly communicating the needs of the client. Using the Agile method, if there were any concerns or changes that the client wanted to address, I could take them to my team and communicate those items and let them get to work. Good communication from the Product Owner is essential, but so is not being micromanaging. Letting the rest of the team focus on their respective roles and their work is essential to good team cohesion. If the Product Owner were to overstep that line into micromanagement, it could lessen moral, decrease productivity, and possibly create an unfriendly work environment.

Another aspect of the Product Owners position is to hold user meetings. Hearing feedback directly from users of the product you are making is an essential part of the development process. This also applies to when the client wants to make changes like I faced during the process of developing the SNHU Travel Project. Taking the information learned from these meetings, I was able to pass that information to the team to instruct them on the changes and answer any questions the team had. I was also able to pass along questions from my team to the client if the need arose. This led to the creation of the backlog with the help of the other members of the team.

**Scrum Master**

The Scrum Master is responsible for essentially managing the team and the tasks that it is working on. Even from a school/exercise standpoint, I can see that the Scrum Master is a vital role in the development process. Handling Sprints, backlog refinement, daily Scrums, and Sprint retrospectives are the vital roles that the Scrum Master is responsible for, and without them, the team will not be successful. In my time in the role as Scrum Master, I believe the most important aspect outside of initially planning the Sprint, was the daily Scrums. Having a meeting every day (regardless of what time) was a way to utilize the knowledge and experience of the team to focus on the important aspects of the project. It also enabled the team to share their successes and seek help on aspects of the project they are unsure on how to complete. The Scrum was also a time for the Scrum Master or Product Owner to share new information or give the team updates on other aspects of the project.

While acting as the Scrum Master, I was tasked to create an agenda for a daily Scrum. Planning out the scrum and having a very matter-of-fact meeting of what each member did, what each member will do, and what problems the member has is effective in keeping the meeting light on time, but also very productive for the entire team. This lets each voice be heard and I as the Scrum Master could take notes and prioritize or assign members to help as needed. If any questions or concerns were brought up that needed addressed post Scrum, that could be scheduled as well.

**Tester**

The product tester is a major role on the team. Not only are the testers responsible for testing the product to ensure its workability and functionality, but they are also responsible for creating the test cases the developers use to create the product. While I was developing the test cases for the SNHU Travel Project, even when the changes came, it was the testers responsibility to create the test cases and make them clear, concise, and understandable. This ensures the developers are aware of what specifically they need to develop. The acceptance criteria field inside those test cases were probably the most beneficial. It listed all the actions that must happen to ensure that area of the product functioned properly.

Additionally, adding the user stories that were given to us by the product owner gave insight that while you have some information, you need to get more. Again, we see communication being a key part to the development process. During one of the scenarios, I had to send an email asking for more information, this is what was sent:

To: John

Subject: User Stories Clarification

After viewing the example that you have sent, I require some clarification on a few matters. For starters, what will the webpage look like and where should the “Top 5 Destinations” link be located? Is it going to be a link that is pressed or a button with the name? Based on the requirement that the Top 5 Destinations is to be based on the individual user and their history/searches, would it be good to have it located in their profile page? The test cases I have previously submitted imply that a user profile is present, this implies the question of what the login page will look like?

Thank you for your time,

Concerned Tester

This example is a good indication of how communication is needed between the different parts of the development team. Feeling able to communicate with your boss or supervisors to get this information allows the test team to ask questions freely and better compose the test cases, especially after getting user feedback and needing to change some of the test cases.

**Developer**

The developers on the Agile team are responsible for the code and structure of the software that is being made. While this is not their only responsibility, that is their focus. Interacting with the testers and ensuring the acceptance criteria of the test cases are met is another focus. One of the main changes for the developer in the Agile environment as opposed to the Waterfall environment is changes to the product during the Sprint. In the Waterfall environment, no changes are made at any time during development, it is saved for after the initial product is developed. Agile allows for changes to be made mid Sprint and can facilitate better results and better satisfaction from the client and users.

During this lesson, I had to deal with a change coming from the client on what content the SNHU Travel Project would provide to the user. I can understand how this can foster a stressful environment, but when the team comes together after these changes are announced, it can be fixed relatively smoothly. Communication with the testers and product owner on the specifics of these changes is paramount. These changes, especially if coming from user stories, are much needed and need to be categorized as important as this will better enhance the relationship with the client. While the developer won’t work directly with the client, it makes our team look better overall when the client sees these new changes implemented properly.

**Agile Phases**

Throughout this course I have learned of the different phases that the Agile method uses during the software development lifecycle. Going through each role and learning the specifics of each one opened my eyes to how a proper team should be built. Not that a team not directly resembling this is ineffective, it is just a good baseline to have when viewing the Agile workflow. For starters, the scrum master’s responsibilities were brand new to me. I knew he ran things, but not how much he was responsible for. Specifically, the responsibility of managing the daily Scrums, sidebar communications, and being the liaison for the team to the product owner if necessary.

As the product owner I saw how important user stories were to the development and test team. Utilizing the product backlog efficiently also holds an important role as it sets the pace for the program. No matter what method is used to characterize the stories (i.e., Planning Poker, Estimation Units, and Affinity Grouping) it is an important step for estimating timelines to report back to the client with. Creating the user stories and product backlog while being the product owner helped me understand how important that role is while assuming the developer and tester role later.

Assuming the role of a tester (even though I do that professionally currently), I was able to learn more about what I do from a more scholarly perspective. For example, the “Expected Results” category I knew was important, but these can be extremely concise and still get the information across. It does not need to be complicated, as that can confuse or mislead the developers without the intent of doing so. Clear and concise writing is sometimes more practical that long drawn-out explanations.

Acting as a developer opened my eyes to the importance of communication and clarification. When the changes occurred in the lesson, we were provided with a script that answered our questions on what was needed. In real life, those clarification need to be done by me, the developer. Reaching out to the product owner and testers with clear intent for the need of clarification is highly important. Also realizing that change is inevitable in the Agile workflow, having an open mind when changes come in can make your life a little bit easier.

Pros and Cons

The Agile Methodology is unique in my mind. I imagine it as a train track being built, and there is a possibility of needing to go around a rock or build a bridge at any moment. Change happens, but in Agile it is almost guaranteed. I think that is one of the pros of Agile, it’s flexibility. When managed correctly, these changes can be properly filtered through the stages, and be distributed evenly before the next update to the product. The Agile method also allows for very instant feedback, as users will see those changes sometimes relatively quickly after an update or change is made.

The downside to Agile is sometimes the time needed to complete the project. If you had a set Sprint for the project, constant changes might push that release date back, due to other things being added in. You might also have a customer that wants more and more changes after seeing their other changes or ideas being implemented in so quickly. I feel that if the product owner and scrum master are not managing the project correctly, this can put much stress on the rest of the team.

**Conclusion**

All in all, I have enjoyed learning about the Agile process. It has helped me learn new aspects about my job that I can implement, and gave me a foundation into something I knew about but didn’t have a full understanding in. Now, I can take these items learned and implement them into my professional life more effectively, and hopefully improve the quality of my test cases as well as my communication with other members of my team.