CS250 7-1 Final Project Retrospective

Brandon Lombard

Southern New Hampshire University

10/18/2020

Retrospective

Over the span of this course, I have taken on the roles of Scrum Master, Product Owner, Tester, and Developer. Each role had differing requirements and workloads, especially when considering which approach that was taken, whether it be waterfall or agile. When I was the Scrum Master in Module Two, I was responsible for keeping the team up to date with project information and changes. Through completing scrum ceremonies including sprint planning, daily meetings, backlog grooming, and spring retrospectives, I was able to keep the team in the loop. To me, the best way of keeping the team up to date was the daily scrum meetings. This is where the team gets together and discusses the current project and progress updates. The meeting typically has three questions: “What progress have I made thus far,” “What will I do today,” and “Are any impediments present that interfere with the project?” These questions are kept simple but are powerful in the project process as it gives the team the information that they will need to adequately finish the project.

In Module Three, I learned thoroughly what being a Product Owner entails. The Product Owner is usually a project’s main stakeholder. As Product Owner, I created user stories and created a product backlog for the team. To do so, it was important to obtain information about the features, functionality, and the system for the software product. Communication with the users is also important because a Product Owner needs a clear understanding of the requirements and user needs. Specifically, in my “3-4 Journal: Product Owner,” I said: “At the end of the day, the individual users are your customers, and as a product owner, it is crucial that communication lines remain open between the customers and you/your team.” Ways of efficient communication can be through note taking, showing respect while using positive language, and proactively taking responsibility. Regarding Agile software development, user stories are one of the important tools that assists in capturing the function from a user’s point of view. With this, descriptions of requirements are simplified due to clear and concise characteristics. After obtaining the required information, I summarized the user’s needs so that the development team can fully understand what needs to be accomplished. After the requests have been completed, I put this information into a product backlog which typically should be ordered by importance level. Finally, the product backlog is discussed by the team while keeping the backlog updated and preparing for upcoming sprints. Being a Product Owner opened my eyes on the scrum process and taught me about keeping information visible for the team and setting clear requirements for them to produce a well-rounded software product.

When I held the position of software tester in Module Four and the group project in the “6-1 Discussion: Vision Quest Software Case Study,” I found that the tester’s job is to use the user stories from the product owner and run test cases to determine whether the requirements have been met or needs updating. Before this course, I thought that the testers would help with the project at the tail end and just test the software, but that has been proven false. They start their work from the beginning of the development in Agile and consistently provide feedback to the team. During the 6-1 Discussion, the team discussed their opinions on what methods should be used and conversed on the project that was being developed. I believe we communicated well by using active listen skills to understand each other’s ideas. For example, most of the team agreed that test driven development was important alongside 15-minute daily meetings. Through this experience, I have found that testers are incredibly important to the development process. I also learned that failure is important in developing software. Failure enhances your problem-solving skills and experience.

When playing the role of developer, it was imperative to have adaptability. In the projects throughout this course, requirements changed, and I had to adapt to meet these requirements. Agile methodology made it easy to adapt in that a core value of agile is to respond to change over following a plan. This makes it easier to make changes where it may be needed. I did not find much trouble making changes to already written code. I typically write considerably basic code with comments to ensure I remember what each piece of code does. This leaves me room for adjustment as agile is great for unexpected issues. Although it may be favored to go completely according to plan in the development process, it is better to make changes where needed. Constantly making changes, however, can lead to running past deadlines and/or writing too many updates that the code may get messy. If the developer remembers their best practices, there should not be an issue.

Conversely to agile development, the waterfall method divides the development process into distinct phases through a sequential design process. Additionally, there are different roles associated with the waterfall method. The Project Manager is the first role regarding the waterfall method, similarly to the Product Owner in agile. The Project Manager sets up a structured plan with deadlines and goals for the team. Then, the Business Analyst oversees creating the approach in the beginning while the Project Manager is responsible for managing the team to meet their goals. The Business Analyst is similar to a Scrum Master in that they work alongside the Project Manager. The developers and testers are mainly similar in both methods, so there isn’t a need to go into much detail. To me, agile development is better because it mainly focuses on team work instead of separate teams, and it gives more room for creativity. In the Waterfall Method, the standards are set and must be completed. In Agile, you have requirements, but you aren’t told exactly how to do it, therefore leaving more creativity to the developers and allowing for flexibility. I prefer having flexibility and more creativity than being told exactly what to do, although it may be more time consuming at times.

This course has given me a great deal of knowledge that I will use in my career as a software engineer. I did not know anything about Scrum events and had no clue what a Scrum Master was. This course material made me feel comfortable with the software development process and made me prefer agile over waterfall. Although I just want to become a software developer, this information is what I needed for my future. Not only did I learn factual information regarding software development, I also learned about concise communication and teamwork. I am grateful that I could use this course to become a better programmer and student.