Matt Knutson

CS-250: Software Development Life Cycles

SNHU - Instructor Farley

April 20th, 2023

Final Project

During the project, the product owner was the first point of contact for the client SNHU Travel. From that point on, she created a backlog and created tasks for the team members, based on that backlog, in the form of stories. The Scrum Master worked directly with the product owner to manage the backlog and make sure the team was on task with the current sprint. The Scrum Master also facilitated the daily scrum meeting and made sure the team stayed on topic during these times. The developers, including myself, took the stories from the sprint and turned them into computer code. We also updated any code, or parts of the running software, that the client decided needed to be changed during the development process. The testers tested our finished code and made sure that it worked as needed. They also help to make sure all of the code blocks work in harmony together. The client, of course, was SNHU Travel.

The Scrum-agile approach requires the stories to be very simple by nature. This allows the project to be broken down into very small pieces that can be created and then finely tuned before moving on to the next task. By hyper focusing on small tasks, work is able to be accomplished faster and more precisely. By documenting the user stories in this project, the team was able to keep track of what was finished, updated, or incomplete. It also allowed the team an

in depth look at what the requirements were for each story. This made it easy to come back and make changes if necessary.

During this project, the client decided to change direction concerning the functionality and design of a particular part of the application. Because we were working in an Scrum-agile environment, we were able to make those changes and demonstrate them to the client during development. If we had been working in a waterfall style environment, we would have not had the opportunity to make these changes until the end of development. If the client wasn't able to be personally involved in the project, and didn't know what changes were needed until the finished product was released, this could cause a lot of issues. The small changes could cause multiple bugs to the running system, or the very least, the client would just be unsatisfied with the final outcome.

During this project, I communicated effectively with the team in several ways. By attending the daily scrum meeting, I was able to collaborate with the team about my current progress and any obstacles I was facing during that time. I gathered information from the team and helped create stories for the current sprint. I also communicated through email with the team about updates I had made to stories and to gather feedback from them about any of my releases or updates.

During the project, we utilized many organizational tools and Scrum principles. We planned our sprint and held daily scrum meetings. We also had sprint reviews and finally a sprint retrospective. During the project we also used burn-down charts and Kanban boards to track our progress during the sprint. All of these elements helped to create a strong team mentality and

workspace. These tools helped facilitate open communication amongst the team and the ability to quickly resolve any issues, or make updates during development.

The pros to using the Scrum-agile approach definitely outweigh the cons. It was kind of inconvenient to not know exactly how long the project could last, due to its flexibility. Also, I could see it becoming a hassle to have to make changes constantly to the original plan, but that actually keeps the process exciting. Other than those two small issues, everything else in the Agile environment seemed much more efficient. I do believe it was the best option for the SNHU project. I love how the Scrum-agile workplace promotes teamwork, and I especially love that it promotes all team members being knowledgeable in all positions on the team. I feel we were much more productive being able to make changes mid-development also, rather than having to wait until the end of production to run the product by the client. I would definitely recommend agile in most complex situations where the final vision is still adaptable and the project can be extremely customizable.