**CS-250 Final Project**

Kayleigh Kinsey

CS-250, Software Development Lifecycle

Professor Morrison

April 11, 2023

Each role contributed something different to the final version of the product. The Scrum Master helped to facilitate conversation between other team members by setting up meetings such as the daily scrum. They mostly stepped back and allowed the team to self organize, only there to provide guidance when needed. The product owner managed the product backlog. This includes creating user stories based on the requirements defined by stakeholders, prioritizing them, and keeping them organized. The product owner also had meetings with the client and the client’s users to determine what they want and don’t want in the final product. Thus, the product owner also acted as a medium between the team and stakeholders, and represented each party to the other. The tester’s job was to take the user stories and write test cases to outline exactly what the program needed to do for the user story to be considered “done”. The developer wrote code for the product, building it to the specifications of the test cases and user stories. As long as the requirements were met, the developer could go about this however they wanted to. Everyone worked together to generate ideas for the final product. They also didn’t stay strictly within their roles, changing hats when necessary to help their teammates.

Scrum-agile is different from the waterfall method because planning, development, and testing are all done simultaneously. The product owner created user stories and gave them to the tester, who determined what the system needed to do and how to test it. The test cases were then given to the developer, who created working software to accomplish the goals of each user story. As this continues, user stories are completed throughout the process, not just towards the end. In the waterfall approach, the user stories (or requirements) are created first, then the project is completely planned out. After that, it is built and tested. Stepping backwards in this process when there is a problem is very difficult, so some of the client’s requirements may not be met, or they may come up with new ones that can’t be included. By following Scrum-agile’s iterative process, my team was able to consistently create, build, and test as user stories developed and changed.

Agile is built around the fact that it is impossible to know all the requirements of most projects from the beginning. Agile practitioners know to expect the unexpected. When the client decided on a wellness theme for their packages, the team was frustrated. We feared we would have to start over, but were reassured that the client’s request could be fulfilled while keeping most of the current code intact. The new theme altered user stories, test cases, and the program itself. Though the change required some backpedaling to implement, there was actually very little that needed to be modified in the end. Since we had received this new requirement early on, the system was still small and didn’t need much adjusting. If we had been using the waterfall method, the product would probably have been much further along when the client decided to change themes. The team would have to make more changes and sift through lots of code. What took the agile team a couple hours to fix could take the waterfall team days.

My team often communicated over email. We tried to write clear questions that conveyed to the recipient exactly what we needed to know. We offered example answers where appropriate to further convey the sort of responses we were expecting, and for problems that needed solutions, we shared our ideas rather than simply asking a teammate what to do. These emails helped everyone to communicate effectively with each other, and ensured that even when my teammates couldn’t talk in person, they were able to answer each other’s questions, contribute thoughts, and generally keep everyone moving forward as a cohesive unit. Though email is not the clearest and most effective way to communicate, the team members tried to write them in a way that discouraged endless back and forth or misunderstandings.

One principle of agile is that the team must work closely with their stakeholders and ensure they are happy with the final product. My team communicated often with end users and the client, taking in all of their suggestions. When the client wanted to make a change to the system, we did as they asked. The team understood how important it was to repeatedly deliver to the client and communicate with them so that the final product is exactly what they need. The main tools that my team used were the scrum events. We had a meeting to groom the product backlog and assign sizes to the user stories. This meeting was useful because the entire team participated and decided together on how to arrange, prioritize, and weigh the user stories. This built trust between team members, established their responsibility for the product, and helped them understand what is ahead. My team also held small retrospectives every week, going over the previous week’s events and considering how the lessons they had learned would be useful going forward. This helped them to slow down and really consider what they did well over the week, what could be improved upon, and what new skills they could take with them into the next week.

I believe that Scrum-agile was the right approach for this particular project. The client knew they wanted a vacation booking system that offered trendy packages, but other than that, they had no certain requirements. This amount of uncertainty for a project of this size doesn’t work well with the waterfall approach. The client would be forced to come up with all the details for their system up front, and any changes they wanted to make could be costly, time consuming, and even impossible. The Scrum-agile approach allowed my team to work fluidly with changing requirements, making adjustments to our work as necessary. We could continuously deliver working iterations of the system to our client, and the client could give us feedback on our work. The client didn’t have to worry about knowing all the details in the beginning, and could view our progress as it was happening, giving them more control over their product. There is a major downside to using the Scrum-agile approach for this project, though. The client stated that they needed the product completed before their busy season, but it is difficult to estimate how long an agile project will take, at least, for the first few sprints. It was difficult to guarantee we could deliver a finished product to the client by the deadline. Still though, I feel like the Scrum-agile approach was the right one for this project. The client was able to watch our team’s progress and continue providing updated requirements. If they knew everything they wanted up front, maybe a waterfall approach would have been a good choice, but since there was so much uncertainty, the client was probably better off with an approach that left room for change. Ultimately, they were more likely to get a product they were happy with because we used Agile.