Alex C. Chadburn

Southern New Hampshire University

CS-250: Software Development Lifecycle

Professor Jasmine

February 22, 2025.

The way our Agile team operated was with various roles, some specific contributions would be the Product Owner for getting user stories, translating them, and backlogging them. The developer developed the specific software that fit the criteria that was found in the backlog, while the tester ensured that it met the criteria and had no edge cases/issues. The Scrum Master facilitated Scrum Events such as the Daily Scrum in order to keep everyone on track.

Because of the Agile model, we get more input from users, and we estimate a larger time, but in the end, everyone gets what they want. Specifically with user stories, the Product Owner first gathers people who want to give feedback, translates the requirements, then puts it on a backlog. There was a user story that wanted to be shown a specific type of trip. Then the Product Owner adds trip filters to the backlog, and it gets done based on priority.

With how the Scrum-Agile approach works, there is extra room for changes and interruptions. Everyone is aligned with their tasks, and the important parts of the project get done first. This ensures at least a good value proposition at the end of each sprint. A specific example would be the developer changing the contents of the SNHU Travel page. They wanted to change the specific type of trips to detox/wellness. Or even the styling of the webpage from being 1 page to being a slideshow.

“Hello Team! I will be the team's tester; I selected this role because I am familiar with technical requirements, test cases and edge cases. As the tester for this project, I will be helping with the transition from the rigid Waterfall methodology to the Agile methodology. Over time with Agile, requirements change, and it is imperative that with that the test cases change, and the overall functionality works with these changes.” This is a specific example, the reason this works so well is because it is direct, addressing some key questions.

Who am I? What do I do? What am I here for? Then I posed questions for myself such as “What can I do to help my team?” This overall helps with communication by being crystal clear and setting expectations. As this is a smaller excerpt it is not seen in as much detail unfortunately.

The Agile principles that helped my team a lot were the Scrum Events. These events help everyone stay on track, identify potential issues, plan for future development, and look at past development to spot any issues or places to improve on. Jira has also helped in many areas such as Sprint Planning and Backlog refinement. It can even help in Sprint Reviews, as you can see specific tasks which have been completed, worked on, or are yet to be started.

The pros would be the wait time is **usually faster**, Waterfall methodology usually struggles and has a backlog of a lot of different companies and in the end, you get something that barely works, if that. **The price is usually less**, they don’t have to spend a lot of time planning for every single piece of a puzzle and pricing it at such. Finally, the product is usually much more **in-line with expectations**, if you asked a Waterfall development firm to change the trip destinations to wellness/detox, you would be out of luck after that phase is over. It would also be difficult even if it was ongoing because of the lack of transparency. Agile might struggle with specific documentation and it can be slightly less predictable, and they might not have a specific vision of the end product.

For such a small project as SNHU Travel was, SNHU itself could have gotten suggestions for their website and contacted a Waterfall firm for maybe a cheaper price. Now this is fine, however, they are in the Travel business, and most businesses are time sensitive. They would have a set date that the firm would be done by, and they could shop around for a faster date. However, 6 months is a bit unreasonable for Waterfall development from what I have heard, so I believe the right choice is Scrum-Agile.

Overall, my time doing Agile in SDLC was pretty fun, getting to play as the different roles, troubleshooting issues, translating requirements and looking through specific strategies such as burn-charts. Some areas I could have improved in would be communication and getting a deeper understanding of each role. It really shows how much I have grown from previous journals until now, where I have a better understanding of what each team member is expected to do.

I can say I have seen Waterfall methodology in place before, especially for meeting people. It is usually fine, but sometimes there are emergencies et cetera, whereas if you have it in a general timeframe, you are much likelier to meet up. This type of experience reinforces my preference for Agile, which offers a more adaptable approach.

From this class, I've learned just how important direct communication is, how these projects are estimated, and what the team dynamics are. These insights will be very useful in future team projects, helping me collaborate more effectively and deliver better results.