

Sprint Retrospective and Review

Mark Webster

During the course of the SNHU project, I took on many roles throughout. As the Product Owner, I was responsible for translating client requests into user stories. Using the client meetings animations that were provided, I was able to synthesize some core desires (the personalization of trips, the “Top Five” features) into smaller artifacts that could be put into the Product Backlog. This role helped me see the importance of client communication and the need for creating user stories that are small enough in nature to be worked into ongoing tasks. For example, the Top Five feature had multiple levels – the user desire to see personalized selection based on previous trips or a determined profile, the ability to see trips only in certain price ranges or popularity. All these features together were an epic, and being able to parse it into smaller tasks for the backlog was crucial down the line for the developers to gauge the work needed.

I also worked as the Scrum Master, the “bridge” between the stratification levels of the project. I was responsible for holding Scrum events, planning sessions, and facilitating communication between the members of the team. I grew to understand that the Scrum Master is the sounding board for the project, able to neutrally transmit information between parties to ensure greater efficiency. The Scrum Master also has to control the narrative at certain points. For example, the animation showed a potential problem with an employee who is constantly late. The Scrum Master has to control the flow of energy to focus on the tasks at hand, putting any other issues into the sidebar to be approached outside of key Scrum events. The late employee can be talked to later in a separate meeting the affected parties; the key is to keep productivity on track. Working with the Product Owner to refine the Product Backlog helps the development team because I am in the loop about their needs and timelines and I can help the Backlog get sorted into a more efficient layout depending on current progress. For example, when we broke the Top Five epic into shorter user stories, I was able to help the Product Owner prioritize them so that essential features were done first with secondary features after.

As the tester, it was my job to design acceptance criteria for given user stories. Perhaps there was a response necessary when interacting with a web element, or a desired layout necessary on certain step. When designing the acceptance criteria, creating steps that a user would take paired with the required result was essential. Testing is all about clarification—clarification of the means of interaction and the subsequent reaction. For example, when I was designing the criteria for the “Sort by Price Range”, when the user selected their price range, the webpage could not show any results outside of this range. If it did, it failed the acceptance criteria. These criteria help the developers as they know exactly what the features they are designing need to do. The tester also needs to be dynamic as the shifting nature of projects often requires change with short notice.

Finally, working as the developer involved planning and execution of user stories to fulfill acceptance criteria. Figuring out which user stories to work on based on estimates of time necessary, the priority of the tasks, and the pace of the project as a whole was tricky. At the end of each Sprint, we want a viable, potentially-shippable feature. This estimation process is a whole-team effort, as is delegating responsibilities within each sprint. Being a developer necessitates flexibility as the Backlog is refined, the client changes directions, and staff demonstrate strengths and weaknesses. For example, we had to edit the jar files to reflect the changes when the client decided to embrace detox/wellness vacations. The development process is dynamic but results in a clear, team-based plan heading into each sprint, and tangible progress along every step.

The Scrum-agile process helped user stories come to fruition through each of team's roles. With the Top Five feature, the Produce Owner was able to take that desire for the client, translate it into the necessary user stories, and prioritize it in the backlog. The Scrum Master was able to assist the development team in planning and figuring out who would do what and how long each feature would take. The testers generated the acceptance criteria for each task, and the developers followed through using all the known information. When the client pivoted their approach to detox/wellness vacations, the Product Owner and Scrum Master were able to get this information to the devs and testers who

could decide on the changes to be made (thankfully pretty straightforward in this case). The quick-react, flexible approach allowed us to change things easily instead of being stolidly stuck in the next phase of development, unable to look back and revise previous work. The daily stand-ups were essential for day-to-day progress checks, and the estimation processes like planning poker helped us decide as a team what would take how long.

The various emails that I sent throughout the course of the project always served a concrete purpose: clear, concise communication. I do not believe in sending emails for the sake of it. Each message should accomplish some forward step in a project, be it clarifying a detail of a project that was unclear before or connecting group members who need to communicate or work together. For example, the Tester email was meant to clarify exactly what the clients were looking for in the profile page feature of the SNHU Travel site so I could generate exact acceptance criteria.

Ultimately, the client-focused approach is what made the Agile approach such a success for the SNHU Travel project. As the client was involved intimately in communication of their needs, we were able to paint a detailed picture of the project as a whole. This allowed clear user stories to be generated, acceptance criteria to be designed, and Sprint to be executed efficiently. When the client changed their minds about things, this could be communicated quickly, and because we had not set a plan in stone, we were able to react quickly and make changes happen. An Agile approach may not be a great fit for a project in which a lot of it is known upfront with little chance of change, but in our case it worked very well because we were starting with higher-level ideas and transforming them into tangible action items as we progressed. Thanks to Agile, we were able to accomplish what the client needed while remaining nimble and efficient.