Brandon Mullins

CS-250

Module 7-1 Final Project

Southern New Hampshire University

Another sprint is wrapped up, and we’ve successfully delivered our product! As part of our routine, we’ll sit down for a Sprint Review and Retrospective to talk about what went well, where we can improve, and how we can keep the momentum going.

As the Scrum Master, my main role was to facilitate communication among team members and ensure the sprint kept moving forward. One of the most effective practices we implemented was our daily stand-up. These meetings gave everyone a chance to voice any concerns, share successes, and discuss obstacles. At first, getting everyone to open was a bit tough, but as we progressed and saw the impact of our efforts, team members started to feel more comfortable sharing their thoughts. We also used a centralized Scrum board, which was visible to all, to keep everyone updated. Plus, we had software tools to track our progress, which was crucial since some team members were working remotely.

Our Product Owner did a fantastic job connecting with clients through regular focus groups. As we developed features, they would bring feedback back to the team for discussion. A significant change occurred when we decided to shift our focus from a generic travel site to one dedicated to wellness retreats. This decision was informed by conversations with potential users, which allowed the Product Owner to compile a list of desired features. The developers collaborated closely with the Product Owner to review these changes before adding them to the backlog, resulting in a swift turnaround that met user expectations.

The Developer Team worked hard to create a solid software product. By teaming up with the Testing Team, they managed to meet all the Product Owner’s requests within the sprint timeline. Testers provided essential feedback that helped the developers identify and fix bugs, ensuring the code was tight. As we updated user stories to incorporate the new wellness features, the developers adapted their expectations and even created test cases for these additions. By the end of the sprint, we were able to deliver a working product to the Product Owner.

Focusing on high-priority user stories allowed the team to tackle the most important tasks first. This approach gave us some breathing room—if the high-priority items took longer than expected, we still had time to adjust. Fortunately, we managed to complete both high- and lower-priority features within the sprint thanks to our Agile methodology. The mid-sprint feedback made it easier to adjust based on user needs.

Looking back at our use of Agile for this project, I can confidently say it was worthwhile. The Agile approach really shone when the Product Owner requested changes based on user feedback. We were able to implement these changes quickly and without losing our stride in the sprint. The ability to modify our product backlog on the fly was a game-changer, especially compared to the waterfall method, where such changes would have to wait until after the final product was delivered.

As Scrum Master, communication was key to keeping everything on track. I made sure to send regular email updates to keep everyone in the loop, and we maintained an information radiator to visualize our overall progress. Daily stand-up allowed team members to voice their thoughts, and I facilitated these discussions to ensure we stayed focused and addressed issues as they arose. These stand-ups also encouraged developers to pair up for more efficient coding sessions. We used a physical board to display sprint goals and accomplishments, which some team members appreciated, while also utilizing online tools for remote participants.

Of course, Agile has its ups and downs. Unlike the waterfall approach, Agile encourages a self-managing team, fostering a sense of ownership and collaboration. However, some people may feel uneasy in an environment without a clear leader. While I provide support, the team is designed to operate independently, which means members need to take charge of their own tasks. Usually, this doesn’t pose a problem; once everyone buys into the Agile process, they can adapt quickly due to strong communication and teamwork. I’m always sharing information, but the team also needs to keep each other updated, which is more effective in a self-directed environment.

In the end, for the SNHU Travel website project, Agile was definitely the right choice. The team was able to pivot swiftly when the Product Owner requested changes, demonstrating the flexibility of the approach. We needed various features, and not all of them were top priority. By breaking them down into user stories, developers could work on them throughout the sprint and get immediate feedback on what worked. This flexibility allowed us to incorporate new features based on user feedback well before the final delivery date. Had we followed a traditional waterfall method, we likely would have faced delays while trying to rework certain aspects. The Agile approach kept us moving forward without sacrificing quality.