**Sprint Review and Retrospective**

Alyssa Elder

Southern New Hampshire University

CS-250 Software Development Lifecycle

Nathan Braun

October 20, 2024

**Sprint Review and Retrospective**

All members of my Scrum-Agile contributed equally to the success of the SNHU Travel Project. Firstly, the Product Owner laid a lot of the major framework of out project. They met with users and stakeholders in focus groups to identify requirements in order to make the User Stories. They were also responsible for creating and updating the Product Backlog, which was especially important when the SNHU Travel management had a shift in their requirements. Next, the Scrum Master was responsible for enforcing and guiding the Scrum practices. This includes supporting the Scrum events like Sprint Planning and the Daily Stand Up. The Tester works closely with the Product Owner. They interrupted the variety of User Stories to develop Test Cases for the developers. This job is essential to a properly functioning program. Lastly, we have the Developers. The Developers are responsible for writing properly functioning and efficient code. In Scrum they make sure that there is a function program complete at the end of each sprint. They also kept up with any changes or shifts in the Product Backlog.

User Stories were very important in creating a product that met and exceeded expectations. The Product Owner started this process by holding a focus group with users and stakeholders. In this meeting the Product Owner asked clear and guided questions about what the users expected from the travel booking program. This included any special search features or functionality tools. They clarified any unclear areas where we needed more information. The Product Owner then used this information to write up the User Stories. The User Stories were then placed into the Product Backlog according to priority of importance. These User Stories were then used by the Tester to create Test Cases that would lay the path for a properly function application. Without these User Stories, we might have made an application that was incomplete or missing features that users desired.

During our work with SNHU Travel, we encountered one shift in expectations. We were told halfway through that we needed to switch all the travel locations to focus on wellness and detox locations. Because we were using the flexible Scrum-Agile process, this wasn’t an issue at all. The Product Owner called a meeting with the Scrum-Agile team to communicate this change. Team members used this meeting to ask questions and let each other know their plan to continue. For example, the Developer claimed that they would update the existing code. The Tester said they would update the Test Cases for the new shift. And the Product Owner said that we would keep the same deadlines, but just rearrange the Product Backlog. Because the Scrum-Agile process allows for flexibility and change, this shift made no negative affects on our team or the quality

During this project, my team and I communicated In a few ways. One being the Daily Stand-Up and other face-to-face meetings. In these meetings, we clearly communicated our own plans as well as asking questions. Such as when we had the meeting about the shift in destination locations. We also used email to communicate. One example is when I was acting as the Tester. I had to send an email to the Product Owner. In this email I asked questions about including ratings and reviews on the Top Five slideshow, including review and confirmation during checkout, a customer support chat, and a frequently asked questions page. When asking these questions, my goal was for the Product Owner to create more User Stories. The Product Owner should ask the questions to another group of users or stakeholders. I made sure my questions were thorough, and not broad. Asking questions in that manner would ensure I could do my job as a Tester to create the best possible program.

We used many Scrum-Agile organizational tools and principles to create a successful program for SNHU Travel. The first is the Daily Stand-Up Scrum event. This was necessary for communication, such as sharing progress and plans, asking for help, or giving updates on the project. Next, iterative sprints were helpful to ensure we were making progress on our backlog. At the end of each sprint, we had a functional application. For example, even back in week three we had a functional yet not completely polished slideshow for the SNHU Travel program. Another tool that was very helpful in our success was the Product Backlog. The Backlog included all of the items that we needed to include in our project. It was especially helpful when the SNHU Travel management made that change a few weeks ago. We were able to rearrange the backlog to adjust for the change, but still get everything done within the application. Lastly, the Test Cases were very helpful in including all the functional processes of the application, such as a working next and back button. Having organized and thorough Test Cases ensures our program works seamlessly for the user.

In a Waterfall lifecycle, the process is very sequential. It follows specific phases starting at requirement analysis, system design, implementation, testing, deployment, and then maintenance  (Tutorialspoint, 2019). This would not be ideal for our work on the SNHU Travel project. For example, in week five when the client informed us of a change. If we were using the Waterfall method, we would have already been in the testing or deployment stages and we would somehow have to return back to the requirement analysis step again. This would cause a major disrupt among the whole team, wasting time and resources. Agile’s flexibility and frequent iterations let us add in the change simply and easily. Waterfall has many other disadvantages that would negatively affect SNHU Travel had we used it. Some being that we wouldn’t create a working product until the end, there’s high amounts of risk and uncertainty, and the in-depth documentation takes a lot of time(Tutorialspoint, 2019). Some advantages of the Waterfall method however are that it is easy to understand and implement, milestones are easily reached, and it can work well for small projects (Tutorialspoint, 2019).

As I have mentioned before, Agile is flexible and encourages change and improvement at all stages of the project. This means that we are always finding ways to improve and deliver the best possible product, without being constrained to a model we created at the very beginning of its lifecycle. Agile is great because it allows for constant customer collaboration, meaning we have an excellent understanding of their expectations (Tutorialspoint, 2019). Many other advantages of Agile include teamwork and cross-training, it requires very few resources, minimal documentation is needed, and it is easy to manage (Tutorialspoint, 2019). Some disadvantages of Agile are that it highly depends on individuals to succeed, adding new team members can be difficult, and there is minimal documentation to look back on (Tutorialspoint, 2019).

References

Tutorialspoint. (2019). *SDLC Waterfall Model*. Www.tutorialspoint.com. https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm