**Sprint Review and Retrospective**

Michael Tekin

Computer Science Department, SNHU

CS 250: Software Development Lifecycle

Tammy Morrison

4/25/21

Each role in a scrum team is essential to the success of an agile project. Beginning with the Scrum Master, it was incredibly helpful to have someone on the team familiar with scrum techniques who knew how to make scrum work for the project. Additionally, the Scrum Master helped by making sure that there were clear start and end times for scrum events like daily standups, that discussions were kept on topic and informative by preparing questions that each team member could answer like “What did I do yesterday?”, “What will I do today?”, and “What is impeding my progress?”, and the scrum master also helped by looking out for ways to improve the team’s efficacy and efficiency.

The product owner assisted the team by working with the client, users, and stakeholders to get clear ideas of what everyone wants and expects. The product owner prepared what they learned from their communications with the clients, users, and stakeholders to deliver it to the team in an organized and easy to digest fashion. Without the product owner, the team wouldn’t have had someone dedicated to presenting the user stories in a clear way that allowed the team to begin working on them in the most effective way. When stakeholders wanted to change the direction of SNHU travels new tool to focus on wellness travel, the product owner was able to deliver that information to us in a clear way.

Our tester was essential in helping the team, and specifically the developers, ensure that code was working and ready for our users. The tester used information from the user stories to put together test cases and reached out to the product owner if they needed more information about what the users wanted. When the team needed to change direction, the tester was ready and able to modify their test cases to help developers with development.

Finally, the developer was key in making sure we had something to show for all of the work that we all did. The developer used information from user stories to develop the code for the project and was ready to pivot when stakeholders decided to take the website in a slightly different direction. When necessary, the developer reached out to the tester for information on test cases that the code must pass and to the product owner for any information that might have been missing from the user stories.

With the Scrum-agile approach, user stories came to completion through sprints where the team agreed on how to break down user stories into story points that could be completed in sprints and what story points could get done during any given sprint. After the planning phase, the team worked on those specific story points during the sprint. If work was completed with time to spare than additional story points could be taken on. If there was not enough time to complete a story point then, during the retrospective phase, the team needed to assess what went wrong, what the points of failure were, and what could be done to avoid this in future sprints. This assessment could happen during any number of scrum events including a daily standup where the team can talk about obstacles in their way or a sprint retrospective.

When the project was interrupted, the Scrum-agile approach supported project completion in a number of ways. Thanks to the fact that we were using an agile approach, development wasn’t in the middle of a huge user story that may have needed to be totally reworked – thus saving us from needing to scrap much of our work. Instead, with our agile approach, we were working on the user stories in parts and we could work the parts that didn’t need to be scrapped into the project as it changed direction. Along with the modular nature of the work done during our agile approach, Scrum also assisted us by making it easy to make sure we were all on the same page. Given that we had a product owner whose job it was to make sure the user stories were clear and accessible to all team members, developers could easily reach out to product owners for any additional information needed to continue development.

“To: Brian

Subject: Revised Test Cases

Dear Brian,

I’m working on developing the code for the changes that were requested in our last meeting. Would you be able to send me your updated test cases as soon as possible?

Thanks,

Mike”

This short email was effective in encouraging collaboration by making sure that the developer and tester were in direct contact with each other and not just speaking during meetings or through some middle man. By being in direct contact, we could minimize the time it takes to get the necessary information to continue development. A key principle in agile is individuals and interactions over processes and tools – being in direct communication with those who can best help us is following agile principles.

“To: Christy

Subject: User Story Clarifications

Dear Christy,

I’m working on developing the test cases for the user stories and I was hoping you might be able to offer a little more detail on what the users might be looking for to make sure that the site will work to the users’ satisfaction. Specifically, for user stories 1 and 2, will we be implementing a sort feature? If so, in what ways should the users be able to sort listings? E.g. by price, popularity, rating, date added, etc.

Thanks,

Mike”

This email was effective at helping get information that was critical to development in a polite and to the point fashion. It encouraged collaboration by getting different parts of the team to work directly with one another to accomplish the goals of the sprint.

Two agile principles are: “Individuals and interactions over processes and tools” and “Responding to change over following a plan” (*SDLC - Agile Model - Tutorialspoint*, n.d.). Focusing on individuals and interactions over processes and tools allowed us to get important information to those who needed it quickly and efficiently. We emailed team members when information was needed in order to facilitate collaboration. Our team used the principle of responding to change over following a plan when it was decided that wellness travel offered a huge opportunity to the business and being able to pivot to that gave us an advantage. Once that was decided, the team was and was able to begin shifting direction in the next sprint. Daily standups were also key to ensuring that we remained on track as we kept other team members updated on our progress and were prepared to inform them of any obstacles that might be in our way. Our user stories template served as a great tool for keeping everyone on the same page on what needed to be completed.

The Scrum-agile approach was effective in helping the SNHU Travel project develop in a way that deliverer the most value that it could. The pros of the Scrum-agile approach include its ability to help a team pivot when necessary. That ability came in handy when the team needed to shift focus towards wellness travel. The Scrum-agile approach also kept the entire team engaged and helped the team stay productive by facilitating close communication and collaboration. Given that pivoting was what the stakeholders ended up wanting, the Scrum-agile approach was definitely the best approach, when compared to something like the waterfall approach. The Scrum-agile approach allowed us to stay productive without wasting time and effort on work that may eventually need to be scrapped in order to shift directions. Cons of the Scrum-agile approach include not having a clear idea of when the project will be complete until a substantial amount of work has been completed but the team did not need a clear deployment date (the stakeholders may have been able to make use of a clear deployment date but that wasn’t expressed to the team). Another con of the Scrum-agile approach is dealing with a slightly more complicated approach when compared to the waterfall approach but, with the help of the scrum master, this was not an issue.

References

*Scrum: A Breathtakingly Brief And Agile Introduction | Agile Learning Labs*. (2021, April 26). Agile Learning Labs. https://agilelearninglabs.com/resources/scrum-introduction/

*SDLC - Agile Model - Tutorialspoint*. (n.d.). Tutorials Point. http://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm