7-1: Sprint Review and Retrospective

Taylor Nedrow

Computer Science Department, Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor Vianelis Martinez

December 15, 2024

In the fast-paced world of modern software development, adaptability and flexibility are key for delivering functional and valuable software that meets client needs. With an emphasis on iterative development and continuous feedback, the Scrum-Agile framework is an effective method of achieving these goals. In this paper, I will cover how the Scrum-Agile framework allowed for success in this term's project, the SNHU Travel project. I will begin by highlighting each individual Scrum role and how they helped contribute to the success of the project. Secondly, I will review how Agile methodology leads to the completion of user stories. Thirdly I will cover how Agile methodology handles interruptions and alterations in vision. Fourthly I will review a segment of well done agile communication. Fifthly, I will explore how Agile principles helped contribute to the success of the project in conjunction with Scrum events. Lastly I will review the efficacy of Agile methodology in this project and conclude if it was the best method for this project.

The Scrum master played a pivotal role by planning what Scrum events to run and how to run them. This included things such as having a daily Scrum, sprint review, and sprint retrospective. Using the daily Scrum as an example it is specified to be 15 minutes long, happens once a day, at the same time, and everyone talks. Each organization can implement Scrum as they see fit, though straying too far from the core principles outlined in the Agile Manifesto could lose the benefits of being Agile.

The product owners' contributions were instrumental in taking user stories and translating them into the product backlog. The product backlog is the source of sprint planning so it's key to Agile methodology. Specifically as the product owner I took spoken user stories such as features or options a user said they would want, and translated them into a list of features to work on. Additionally, these features were prioritized by importance to the vision so the team would know what sequence to work on items in.

The tester added significant value by helping elucidate the exact pass-fail cases of each user story. Being able to determine how each individual user story is meant to behave is key to delivering functional software. This can be seen in the assignment 4-2 where I took user stories and translated them into test cases. These test cases were then refactored based on product owner input to more accurately reflect client demand.

The developer helped contribute by updating existing code to a new product vision. This included changing resources, such as travel locations, images, and descriptions, from a general focus on travel to a focus on specifically detox and wellness travel. In addition, I did some minor bug fixes, namely changing the way the UI elements were laid out to fix a bug with windows not resizing properly. This was in keeping with delivering functional code.

Agile development helps bring user stories to life by translating them into the product backlog. Then the product backlog is used for sprint planning. Once the user story is selected as the next sprint the developers and testers begin work. All throughout this process the user story can change which it did in this course. An example from this course would be the top destinations list. This user story started as a request from a user. It was then translated by the product owner into the product backlog. The tester then translated the user story into a test case for the developer and got further direction from the product owner on the exact behavior of the software. Next the developer implemented the software functionality for the top destination's slideshow such as a display for the images and text description.

Handling interruptions is a core aspect of Agile methodology. Not only do changes not disrupt an Agile project they improve it by making the delivered product more in line with what the client actually wants. The irritative and short nature of sprints helps this. An example of adapting to client demand was when the project refocused from general travel to detox and wellness travel. The team was able to quickly shift the current user story from general top destinations to detox and wellness destinations. This was started by the client who specified their new vision. Then the product owner communicated this to the team. After this the team made the prerequisite changes to the software.

An example of effective communication between the team and I is from the 4-3 Journal for the tester role. In this assignment, I wrote an email as the tester to the product owner. I used exact language to ask about edge test cases within the software. The text is “how do we handle fringe cases such as, when zero is entered, when a negative number is entered, when a non-numeric character is entered”. These questions were direct, clear, and had as little ambiguity as possible, ensuring that the product owner could easily understand and respond. They were effective in this case to elicit a response with little confusion. A face-to-face contact, such as in person or Team's call, would be preferable but obviously impossible in this instance. Despite the lack of face-to-face communication, this was an effective exchange.

The organizational tools and Scrum-Agile principles that helped my team be successful included “Responding to change over following a plan” (Agile Alliance, 2001). This can be seen in the previously mentioned change from general travel to wellness and detox travel. Following this principle allowed the project to fit the client's vision. The daily Scrum was key to the success of this strategy as it allowed the whole team to be updated about the changes at once and stay on the same page. Without regular daily Scrums the whole team could be working on an outdated vision as management tried to contact each individual team member.

The pros and cons of the Scrum-Agile process that presented themselves during the SNHU Travel project were numerous. The biggest advantage was Agile’s ability to quickly adapt to user input. When the client shifted their vision from general travel to detox and wellness the team was able to pivot on a dime changing out their goals that very sprint. Waterfall methodology would not have been nearly as able to quickly adapt. Now it could be argued that the open communication channel between client and Agile team sets a bad precedent. It makes it so the client can be less clear on their vision and gives them room to change their mind. My counter argument to this is that growing and altering the vision alongside the client is not a con. The Agile Manifesto highlights “customer collaboration over contract negotiation” (Agile Alliance, 2001). Using this method, we deliver value to the client instead of what we think they asked for months or years ago.

I do believe that Scrum-Agile was the best approach for this project. The client altered the vision, and we were able to pivot alongside them. If we were using the waterfall method, we would have been locked into delivering the previous request. With Scrum, however, we were able to change mid-sprint to better suit the client's needs. Even if we had finished the sprint, it still would have been a substantially shorter turnaround than waterfall. This exercise was very informative and only the beginning of a long process of development. I think using Agile on this project was beneficial in the time I got to work on it. If the project were to continue, I see Agile providing dividends throughout. There is no doubt Scrum-Agile was the ideal choice for this project.

References

Agile Alliance. (2001). *Manifesto for Agile Software Development*. Agile Manifesto. [https://Agilemanifesto.org/](https://agilemanifesto.org/)