Sprint Review and Retrospective

Tyler Coplan

Southern New Hampshire University

6/16/2021

The first Sprint on our SNHU Travel project has been completed. The Sprint Review and Sprint Retrospective have been combined in this report to provide examples of the work done and draw conclusions of the process all in the same place. There are three main roles in a Scrum team, the Scrum Master, the Product Owner, and the Development Team, which consists of developers and testers. This report will look back at the first Sprint and give examples of how each role helped contribute to the project. It will also analyze how the Scrum processes and Agile principals facilitated the workflow of the Sprint. There are several tools that the Scrum team utilized to facilitate communication and workflow that will be evaluated here. There were some advantages and disadvantages to the process over a traditional waterfall approach to developing software that will be discussed in this report. Overall, it was a successful Sprint that saw communication and teamwork propel the project past obstacles and navigate through changing requirements.

The Scrum Master role was essential to our process. The Scrum Master was responsible for Product Backlog management with the Product Owner, coaching the Development Team on Agile practices and removing impediments from the Development Team’s progress. One example of the Scrum Master assisting the Product Owner was the initial client meeting for defining the framework of the project. The Product Owner and Scrum Master met with the client and, after listening to her ideas, the Scrum Master came up with technical requirements to meet the client’s needs. This allowed the Product Owner to get started on the Product Backlog based off the technical requirements outlined by the Scrum Master. One example of the Scrum Master assisting the Development Team was during the changing requirements. The Product Owner informed the Development Team that requirements were changing. The Scrum Master listened to the exchange between the Product Owner and Development Team, then asked some questions at the end of the conversation that helped clarify details about the project and deadline. The ability to listen to the team and facilitate communication to make sure everyone is on the same page helps improve the efficiency of the team by making sure everyone has all the necessary details. These examples are in addition to organizing the standard Scrum events, which will be examined at a later point.

The Product Owner played a crucial role in the development of the Product. She was responsible for managing the Product Backlog, defining user stories and working with client on business requirements. One example of her role helping the development of the product was when she held a meeting with end users to discuss product features. She listened to the users and determined what they were looking for in the product to come up with user stories that outlined new product features and functions. These user stories allow the Development Team to develop and implement the software needed to carry them out, as well as estimate their completion time based on story points. Creating detailed user stories and prioritizing the backlog helps the team understand what work needs to be done and how important each part is.

The Development Team consists of developers and testers who each played essential roles in developing the product. Testers were responsible for coming up with test cases for each user story and making sure the solutions developed by the development team met the requirements. Developers used the test cases developed by the testers and the details of the user stories to develop the features in the product. One example that highlights both of their roles was when they responded to changing requirements. The tester was able to get to work developing new test cases to meet the changing product requirements while the developer looked at the current progress of the project to see what kind of impact the changes would have on the progress of the project. The new test cases assured that quality software was being developed and the estimations given by the developers gave the Scrum Master and Product Owner some direction as to how to reprioritize the Product Backlog. The developers and tester worked closely together as a team under the Scrum framework to develop a quality product.

The Scrum framework uses Agile principals to guide a standard set of events to complete work. The Sprint is the main idea behind Scrum framework and sets a timeframe for which the Sprint’s goal is to be completed. The Sprint Planning event is held by all members of the Sprint team to decide which user stories will be completed during the Sprint. The Daily Scrums are held with the Development Team and Scrum Master and it keeps a daily line of communication open where the team shares the work it has completed, the next tasks that will be worked on, and anything that is slowing down their progress. The Sprint Review is held by the Product Owner and Scrum Master and is given to product stakeholders for a review of the work completed during the Sprint. The review may require some developers or testers to demonstrate new features or other things, so they are also occasionally required at these events. The last event is the Sprint Retrospective where the Scrum Master and Development Team analyze the processes of the previous Sprint and find ways to improve the process heading into the next Sprint. These events are timeboxed so everyone knows exactly how long they will take to accomplish. The Scrum framework facilitates the necessary communication between the team which allows the team to be more efficient.

There are different tools that Scrum teams use that aides communication or improves workflow. Information radiators, such as story boards, can be physical or digital displays of information on the current status of the project. It will show what work is in progress, what work needs to be completed and what work has been accomplished. Other informational tools such as burn-down and burn-up charts can give everyone the information about the progress of the project. There are also automation tools, such as JIRA, that not only can provide digital information radiators, but integrate them with automated regression testing. Since testing is done alongside development in Agile, any changes in requirements need to test existing features to make sure that the new features did not break the existing features. These tools are essential to Agile development because they improve communication and testing would be impossible to implement without automated testing.

Agile and Scrum provided some distinct advantages over the traditional waterfall development approach. Agile development allowed our team to get started developing the new product with limited information about the project requirements. This is not possible with waterfall development because the project requirements must be well-defined prior to the start of development. Agile also allowed for changing requirements which was necessary for our product, and was expected because it was a new idea we were implementing with many unknowns. Requirement changes are not welcomed in a traditional development approach and would have led to a product that was missing or had incorrect features. One downside is estimation in Agile practices. Since change is welcome, it can be very difficult to provide an accurate estimation of the timeline for the project. This is easier and more accurate in traditional development since the requirements are defined upfront and not allowed to change once the development has started. The advantages of Agile far outweigh the disadvantages, especially for a new product like SNHU Travel. It allowed our team to get to work on a product, defined the roles each team member and gave us the tools to handle changing requirements. This allowed the team to deliver a product that was inline with customer and business needs.