CS250 Sprint Review and Retrospective

Joshua A. Battaglia

Southern New Hampshire University

Table of Contents

[CS250 Sprint Review and Retrospective 3](#_Toc95493478)

[A. Demonstrate how the various roles on your Scrum-agile Team specifically contributed to the success of the SNHU Travel project. 3](#_Toc95493479)

[B. Describe how a Scrum-agile approach to the SDLC helped each of the user stories come to completion. 3](#_Toc95493480)

[C. Describe how a Scrum-agile approach supported project completion when the project was interrupted and changed direction.](#_Toc95493481) 4

[D. Demonstrate your ability to communicate effectively with your team by providing samples of your communication.](#_Toc95493482) 4

[E. Evaluate the organizational tools and Scrum-agile principles that helped your team be successful.](#_Toc95493483) 5

[F. Assess the effectiveness of the Scrum-agile approach for the SNHU Travel project.](#_Toc95493484) 5

[Describe the pros and cons that the Scrum-agile approach presented during the project.](#_Toc95493485) 5

[Determine whether or not a Scrum-agile approach was the best approach for the SNHU Travel development project.](#_Toc95493486) 6

[References](#_Toc95493487) 7

CS250 Sprint Review and Retrospective

There are various roles in a Scrum-agile Team, with a unique yet cooperative contribution to the success of a project. Some of these roles include the Scrum Master, Product Owner, Developers, and Testers. The Scrum Master’s role is similar to a coach. They assist with backlog groomer, consistently communicate with the Product Owner, track daily stand-up meetings, and facilitate estimations. The coaching aspect comes from encouraging the sharing of knowledge with the rest of the team. The Product Owner is the bridge between the team and the customer’s vision. Defining the backlog, evaluating the teams' velocity, and looking for the most efficient ways to ensure an accurate product for the customer, based on their vision. The developers are the work-horses looking to accomplish the user stories created by the Product Owner based on current priorities, and the Testers create and execute test cases that “QA” the product being developed by the development team (Overeem, 2016).

User stories are a great way to stay organized and accomplish current priorities. The user stories are created based on many perspectives of the customer and customer base. Once defined the team executes sprints to develop all the requirements included in those user stories. A dedicated team works extraordinarily well because their focus can be maintained on finalizing that individual product before release. Once a sprint is completed, the team can regroup and evaluate what was accomplished, what challenges they faced, and lessons learned so they can improve for the next sprint, which in turn will hopefully become more efficient. All these factors contribute to developing a quality product, and the main track to a swift completion without fear of revisiting planning, or failure. Every member of the team is engaged including the customer to ensure the quality and accuracy of the requirements.

Another benefit of the Scrum-agile approach is its flexibility. When the team needed to change priorities or objectives, the agile environment ensured swift communication with the rest of the team. Once addressed, the team was able to discuss the change’s feasibility and expected efforts. Eventually, everyone was able to agree on what was expected for the product to stay on track, despite the interruption and efficiently communicate their tasks and projections. The agile approach ensured this communication and supported the changes by including this shift into the next sprint, rather than scrambling right then and there. The tester would start adjusting test cases, and developers would switch focus uninterrupted, after the meeting.

Once the team moved past the initial interruption, and switched focus to the new priorities, of course, some additional communication was needed to clarify the main objectives and how to accomplish them. An example would be from the tester, asking for software specifications so that they could adjust the new test cases and not slow down the development. This email read:

“Good morning team!

I have recently started drafting test cases for the upcoming software updates/development, and I wanted to gather some amplifying information. Tracking the prioritized user stories, who will be using the upcoming update? How does the development team plan on implementing this feature? Specifically, the price limit user story for the travel agent profiles seems a bit vague. Also, if you had to put a time frame to it, what is your best guess ETA? Attached you will see a template of my test cases so you can see why I need this information. I want to mitigate re-work if possible before myself and the rest of the test team get too deep into the weeds, just to find out everything changed. Let me know if you have any questions about how we design our test cases, and hopefully hear back from you both soon.” (Battaglia, 2022).

Email proved to be an efficient way to communicate with the agile team, especially the developers since it doesn’t immediately disrupt their current process.

Although face-to-face communication is preferred according to the Agile Manifesto, that does not appear to mean all other tools are forbidden. Throughout the development process, the team did agree to use JIRA after careful consideration. This tool is excellent for tracking progress, user stories, bugs, and test results when it is carefully maintained. Eventually, the team was able to discover a balance between tools and agile principles to facilitate productivity and clear communication. Emails worked well during times when developers were heavily working on user stories during a sprint, JIRA was the information radiator, and the Product Owners whiteboard was hung where the entire team could see the current week’s key items. The balance found in agile methodology and available tools certainly contributed to the team’s overall success.

When we assess the effectiveness of the Scrum-agile approach we begin to notice its pros and cons. During the SNHU travel project, the agile approach was flexible and adaptable when changes occurred. Not only did it allow the team to communicate the changes but also allowed the team to self-assess their predicted efforts. The downside to this flexibility occurs when changes get implemented last minute and the team has to adjust before clear information is conveyed by the customer. The upside is creative freedom, but the downside to this creative freedom is the probability of rework or further changes. This could leave the team with uncertainties for an idealistic amount of time.

Upon completing the SNHU Travel development project it was apparent that the Scrum-agile approach was an outstanding choice. The flexibility and creative freedom allowed the developers to accomplish large, high-priority, user stories and tasks without feeling utterly defeated. The testers had an opportunity to assess and evaluate whether the last-minute changes would cause their test cases to delay productivity, and the Scrum Master and Product Owner were able to communicate with the team extremely efficiently. Overall productivity was maintained and the final product was what the customer envisioned. If this effort was completed in a traditional Waterfall method, the changes would have created more delay and uncertainty. Additionally, I do not this the changes and challenges the Scrum team faced would have gotten communicated as swiftly as they were and the product might have failed. From both flaws in communication, and executive involvement.

References

Battaglia, J. A. (2022). Journal 4-3 [Unpublished paper]. Pages 2 – 3. Computer Science Department, Southern New Hampshire University.

Overeem, B. (2016). Characteristics of a Great Scrum Team*.* [*http://www.infoq.com/articles/leadership-challenge*](http://www.infoq.com/articles/leadership-challenge)