Sprint Review and Retrospective

Ian Pruter

Southern New Hampshire University

October 15, 2020

**Agile Roles**

In Agile, the varying roles all come with more liberty and responsibility than traditional approaches. As the Scrum Master, my job was to facilitate meetings and ensure the Scrum team adhered to Scrum principles and best practices. In the SNHU Travel project, I helped with prioritizing and estimating user stories. I started by making a team charter we all contributed to, to ensure the team understood how we would conduct ourselves and communicate. I facilitated meetings like the Sprint planning and daily stand ups. I ensured that we adhered to the topic and time frame. This ensured we maximized time and didn’t overburden attendees with information that would be more relevant in a different time box. During the Sprint planning, I helped the development team and Product Owner negotiate what stories went into the Sprint. During our daily stand ups, I liked to lead with three questions: what did I do yesterday, what would I do today, and what our my obstacles. When I wasn’t in meetings, I would check in with the team and see if I could help out with any tools or skills that they had questions about. I would also handle any issues with the work environment or any inquiries from other departments that could be distracting.

The Product Owner holds meetings with stakeholders and grooms the product backlog. The Product Owner met with the client on multiple occasions to see what they needed from the product. The Product Owner took note of the needs and turned them into user stories. She also met with some users of competitor sites to see what features made the product stand out to them. The ideas inspired user stories that established who the user was, what they wanted, and what it will do for them. That format helps the developers better understand the needs they were developing a solution for. The Product Owner then organized all of the stories by relevancy into the product backlog. This told the team what needed to be done early and what they could try and fit in later, or discard if needed. During our Sprint planning, she negotiated all of the user stories she would like done in that Sprint. As the client’s research suggested new needs from the product, the Product Owner met with them to discuss the new direction of the project. After the meeting, the project owner reprioritized the product backlog.

The tester worked closely with the product owner to define acceptance criteria and establish the definition of done. Like the Product Owner, he also kept in contact with the client to discuss requirements. He then made test cases for each use case and made sure the user inputs gave the expected results. He also discussed the testing process with the developer so she would understand how her work is tested.

The developer and tester figured out the increments of the user stories they would develop and test. The developer wrote a file that displays a ranked list of most popular destinations. She didn’t get a chance to make it into an app and web page before the client’s needs changed. She made some adjustments to the product in the next iteration to change the format to multiple pages ranking detox spas. We all attended daily stand ups where we self-organized to best accomplish the Sprint.

**Software Development Life Cycle**

We learned the client needed a booking system of trendy tourist destinations. This gave us a rough idea of the size of the project and the resources needed. The focus group we held later is where we got the inspiration for our user stories. A list, filters, and customizations were direct suggestions from the attendees. The Product Owner used the information from the meetings to develop the product backlog. In the Sprint planning, we settled on three stories. One story for a top five most popular destinations list, another story for content details and filters, and another story for a cellphone app. We prioritized these stories in the backlog starting at the list, the filters and layout, then the app.

The team broke down the Sprint into iterations to make testing and product changes more manageable. We held daily stand ups to make sure everything was going smoothly and identify obstacles. At the end of every iteration, the tester ensured the acceptance criteria was met. We also showcased our product to the client to make sure we were on the right track. The list ended up going through several reworks where more features were added until the site was both functional and aesthetically pleasing. The filters and details made up a relatively shorter story and did not require as many iterations. There was some overlap with the esthetics with the first story that changed over the iterations. The app was the largest story by far. It didn’t undergo much rework since most of the changes requested by the client occurred in the first story. The app did go through a lot more iterations to convert the computer-based inputs to phone inputs. The GIU also needed some adjustments to work with the smaller dimensions of the two orientations of the phone screen. After some reworks based on change in client needs, we successfully finished the Sprint with a working product. All three features were shippable and the product as a whole, is ready to have new features added to it. We finished the Sprint with a Sprint review and where the Product Owner approved of the Sprint. We then held a retrospective that I’m reporting on now, were we discussed what went well and what we would need to improve in our process for the next Sprint to go over better.

**Mid-Project Changes**

After we had developed a working script displaying the top trending attractions, the client wanted us to make changes based on new information. A couple Agile concepts made this change easier on us. The concept of stories and backlog grooming ensured we had only completed the fundamental features of the product. This meant that when the client had us change the content to detox spas and the page layout, we didn’t have to scrap any coded features. We only had one story we needed to edit. The developer left the container layout the same but split the list into five separate pages and removed the scroll bar. The structure if iterations of the working code meant these changes wouldn’t cause issues with other features. This is because other features won’t be integrated until the current iteration is tested.

**Team Communication**

Sometimes team members had key questions for them to proceed with their work. If a question arose outside of a meeting and wasn’t addressed in the tools, we communicated in the form of email. The emails allowed us to ask a list of specific questions that we could format in such a way that each point would be noticed.

**Tester Email Format Example:**

**User Story One**

* Is the sign in form on the page or is it a pop up window after clicking “sign in”?
* Is the password form under the sign in form or on the next page?
* Can I get some sample data to test with? If could get a matrix or the vectors of the locations of my “previous visits”, I can confirm the accuracy.
* What container on the page does the link go in?

The topics the questions concerned were listed in bold font. Beneath the topic were the bulleted questions. This provided context to the reader and then asked questions to address one by one. A lot of emails included multiple team members encouraging collaboration for anyone the questions would concern. This was a success because it allowed us to avoid any misunderstandings. We all knew what our current understandings of the client’s needs were. With everyone on the same page, our feature integrating, testing, and product demonstrations went much smoother than they could have been.

**Agile Principles and Tools**

The principles of Agile were a huge contributor to our success. We prioritized individuals and interactions over processes and tools. We also made sure to meet up regularly and adapt to changes and roadblocks, as appose to following ridged processes. The tools were great but our in-person meetings allowed for more transparency with identifying issues and reorganizing in a way to best overcome them. We also focused on testing the product regularly over documenting progress, changes, and defects. As mentioned previously, we incremented the product to ensure we met the five-week deadline. We were also transparent with them to positively set expectations. We also adapted to the changes as appose to following our original plan by using the iterative development structure.

Although tools were not the top priority, they were still necessary for our success. We primarily used an information radiator for transparency and a program called Jira for better communication. The Product Owner updated the new user stories on both tools right after grooming the back log. Before every daily stand up, the team would update the information radiator to reflect the current stage of the user story they were working on. I would frequently change the burn charts on the information radiator based on the progress established during the stand up. The nice thing about Jira was the team could update them upon completion of stories. This was a great way for the Product Owner to also stay in touch with our progress.

**Scrum-Agile Approach**

The benefit of Scrum was our ability to adapt when the customer had us make changes to the project. Because all features were broken into stories, we didn’t have to scrap coded features that were no longer relevant. The Product Owner had to reprioritize the conceptual ideas on the backlog. The downside was it was a relatively small project. Some of the meetings with the client and our team may not have been necessary as the progress of the product was steady, for the most part. It also calls for additional positions such as mine that don’t directly contribute to the program itself. Ultimately, I think it was the best approach. The meetings were concise and ultimately didn’t take up that much time. Because of our demonstrations and close communications, the customer knew what to expect and could request changes without causing major disruptions to the process. A more planned approach could have still given them a good product, but Agile gave everyone the assurance that the product was what the customer wanted and that the deadlines were met.

**References**

Eby, K. (2016, July 29). Comprehensive Guide to the Agile Manifesto. Retrieved from <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>

Wilson, F. (2018, October 29). Agile Best Practices Every Agile Team Should Have in Place. Retrieved from https://www.ntaskmanager.com/blog/agile-best-practices/

Ghahrai, A. (2017, January/February). Agile Testing Mindset and the Role of the Agile Tester. Retrieved from https://devqa.io/agile-testing-mindset-tester-role-agile-team/

Gurendo, D. (2015, June 30). Software Development Life Cycle (SDLC). Scrum Model Step by Step. Retrieved from https://xbsoftware.com/blog/software-development-life-cycle-sdlc-scrum-step-step/#:~:text=Software%20Development%20Life%20Cycle%20(SDLC,Scrum%20Model%20Step%20by%20Step&text=Unlike%20the%20waterfall%20software%20development,ready%2Dto%2Duse%20product.