Sprint Review and Retrospective

Through this sprint, the Scrum team, which consisted of the Product Owner, Developers, testers, and me, the Scrum Master, were able to complete the user stories that had been adopted for the sprint. I will now review the contributions that each of these roles played in the process while completing the work on the SNHU Travel Booking Site, as well as describing how the Scrum agile approach ensured the success of the project through these steps.

First, I want to cover the events that were completed on the SNHU Travel Booking site while detailing the contribution and successes of the team members. Those events, or the framework of Scrum, were the sprint planning, the daily scrum, and the sprint review. The first event was the sprint planning event, which was built from the initial client meeting at which the client provided their product goal to the Product Owner and the Scrum Master. After this initial discussion, the product owner and scrum master developed the Agile team charter. This communication device helps direct the team on what they will be working on, how they will work as a group, and the values that they will uphold. Once this had been completed, the product owner held a focus group review with some of the top clients/users of the SNHU Travel site to get their input into the needs of the new site functionality. By capturing this feedback, the product owner could develop simple, easy-to-utilize user stories and assign them based on priority while ensuring that each met the client’s needs. This key role ensured that the agile team understood the goals of the project and the work they were completing through these user stories.

We then moved into the second event, daily scrum, of the scrum framework, the period in a sprint when the work on the user stories, or artifacts, begins. Once the user stories had been developed and assigned to the sprint, the testers took those stories and developed test cases. These test cases were the guiding light for the development team when completing the work. It helped the developers understand exactly what each component needed to do and what the results should be. They were very collaborative, working closely with the product owner to address any concerns or questions they might have had on the user stories. For example, here is a snippet from one of those communications.

*“I need some additional information and details to develop more specific requirements to clearly finalize my test cases. Please review the below questions and provide answers:*

*User Story One:*

*What criteria should the top 5 trips based on previous travel be derived from? Should it be Location, Price, Length, type of trip?*

*If it should be based on all the above, is there an order of hierarchy that should be followed, as in, location, then price, then type of travel?*

*Should this list be narrowed down to the top destinations prior to narrowing down by previous travel?”*

By being clear and direct with their questions, they were able to indicate exactly what they needed from the product owner to finalize the test cases and ensure the sprint was successful.

Once the test cases had been developed, the development team began building out the individual user stories. When completing the work, they were diligent in updating documentation and comments within the code to ensure that all users understood their updates and why they were being made. They communicated directly with the team during the daily standups, informing the group on what they completed the day before, what they would be focusing on that day, and sharing any roadblocks or issues they might be having.

In the middle of the sprint, there was a change request to the end goal provided by the client for the project. This change required a fluid approach to the work, meaning that each member had to provide efficient updates to their work to ensure that the team could be successful. First, the product owner got the team together to discuss the updates requested. They provided detailed information to the team so they could provide their perspectives on whether the changes could be accommodated on the current timeline. Once confirmation had been received, the product owner updated the user stories, testers updated the test cases, and the developers updated the code. Utilizing strong communication between the different members of the team, they were able to adapt and move forward. The Scrum framework was built to support large projects that were not well defined, knowing that work would need to fluctuate based on new information and evolving user requirements. As those changes occurred, the team was self-sufficient and organized, addressing questions and concerns with each other to ensure success. Because user stories in Scrum are small incremental bodies of work, when changes occur, it is easier and more efficient to move forward with those changes. For example, the developers needed to ensure that the changes were incorporated and captured correctly, so detailed emails were sent to the product owner and testers to get the information and answers they needed in a timely manner. Here is a snippet for example.

*“After reviewing the new plan to focus on detox/wellness travel, I have some follow up questions to ensure accurate development of the user stories. Please see below:*

***@Product Owner:***

* *Now that the focus of the slide show is on Detox/Wellness, should the slide show title be updated to reflect “Top 5 Detox and Wellness Destinations”? If not, should it be something else?*
* *With the list now focused on Detox/Wellness, should we still number the top 5, as in “#1 Destination” or should it just be the destination location and short description?*

***@Tester:***

* *With the change in plans, have testing cases been updated? If so, please share so that I can take the testing requirements into account.*

*If either of you have any questions, please let me know.”*

Utilizing the strengths of the Scrum approach, the team was able to quickly react to the changes, and even with the changes, the team was able to successfully complete the user stories prior to the end of the sprint, making the project successful.

Some of the key organizational tools that the team utilized throughout the project included the Team Charter, the User Stories Template, Daily Scrum, the Test Case Template, and email communications. Examples of these have been presented throughout this retrospective review and I will now highlight how these effectively supported the project. These tools were the backbone of the team's success, helping promote the principles that the team had agreed upon at the beginning of the project. During the sprint planning phase, the Team Charter and User Story Template were successful tools to educate the team on the goal of the project, aligning the team’s values and principles, and providing guidance on the work to be completed. The team charter focused on the goal of the project and highlighted the key principles of agile, directing transparency, collaboration, and constructive communication which were critical in ensuring the team’s success. The user stories are one of the focal points of the agile-scrum framework. These small, direct work chunks, developed in a user story template, successfully guided the team in their daily scrum work throughout the sprint, which was the backbone to the completion of the project.

During the sprint, a Daily Scrum was utilized, a daily standup meeting that involved the entire team at which each member of the team provided three updates, what work they completed the day before, the work they would be completing that day, and any roadblocks or issues they are experiencing. These Daily Scrums ensured that the sprint was moving forward effectively and continued to foster transparent communication between the team members and facilitated collaboration in the work. The Test Case Template, which highlighted the user case tests that needed to pass to ensure a done status of each of the user stories, was concise and effective for the developers to be able to act upon. Leaning into the agile principle of prediction and adaptation, the team was able to adopt the user test cases based on the new direction and information provided by the product owner to ensure that the user stories were successfully completed.

Throughout the sprint, effective email communication between team members was utilized to help capture feedback, answer questions, and proved the importance of transparency and collaboration. Examples of these email communications were provided earlier in this retrospective.

When reviewing this project and the use of the agile-scrum framework rather than a waterfall approach, I believe it is clear why the agile approach was the correct option. When working with a waterfall approach, a project is fully defined and planned prior to going into the work. Once the work begins, it is very rigid in its plan, which each section of work being isolated. For example, if this effort followed the waterfall approach, the entire project would have been planned out wholistically, rather than individual user stories. It would have then been passed off to the developers to build the entire program without testing being implemented until they were complete. Testing would have then been done and developers would have to go back and make corrections based on findings. If changes were to arise, as they did, the entire project would need to adjust to reflect it. In contrast, within the agile-scrum framework, the project does not need to be fully defined prior to beginning the work. User stories are developed to support the individual needs of the users/stakeholders, and work is completed incrementally based on the priority level of the user stories, with testing being done concurrently with the programming. This allows for quick and nimble shifts based on updates and new information provided by the user/stakeholders. There can be drawbacks to this approach though, for example, the scrum-agile framework demands self-organized and collaborative teams. If this does not occur, the project could fail. It can also form what is called scope creep. Because it is built on the idea of adaptive and fluctuating needs, constant changes to the plan could cause work to never fully reach the status of done as it is always changing.

While these concerns are apparent with the scrum-agile approach, the team was successful in executing the project, and due to great collaboration, transparency, and holding to the core values and principles, they were able to thrive in their work.

References

Cobb, C. G. (2015) *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. Hoboken : Wiley.

Schwaber, K., & Sutherland, J. (2020) *The 2020 scrum GUIDETM.* Scrum Guide | Scrum Guides. Retrieved February 19, 2023, from https://scrumguides.org/scrum-guide.html