Adam Sissoko

September 1, 2022

Sprint Review and Retrospective

Another sprint completed, and another successful product delivery! As is customary, the team will sit down for a Sprint review and retrospective. During this time we will discuss what worked well, what we can improve on, and what we plan to do next to keep this momentum going.

As Scrum Master, my role was simple: facilitate effective communication between all members of the team and keep the sprint moving forward. Some of the most helpful practices that were implemented were daily stand-ups with the team, allowing everyone a chance to voice their concerns about issues and impediments, and to communicate successes and accomplishments to everyone else. Keeping the team talking openly was a challenge at first, but as the sprint progressed and the team made progress towards our goal, everyone felt more comfortable expressing their opinions and frustrations and excitement to the group. Keeping everyone informed outside of the standup was accomplished with a centralized Scrum board that was visible to everyone. Additionally, the team used software to electronically track their progress long-distance, as there were other teams in far away locations working on the same project.

Our Product Owner’s goal of gathering information from clients of our product was done through regular focus groups and meetings. As features were developed, the product owner reviewed them with the rest of the team and we all discussed what could be done better and what worked well. Sometimes the features needed to be modified based on current trends and changing goals. This was when the product owner met with developers directly to discuss these changes. One such change was the pivot from a generic travel website to a website that focused on wellness retreats and vacations. This change was discussed with potential users first, and then the product owner came back with a list of features they would like to see implemented in the new design. The developers and the product owner took the time to review these changes together before making the addition to the backlog, resulting in a quick turnaround and a final product that met all expectations.

The Developer team worked tirelessly to accomplish the goal of creating a solid piece of software. Working closely with the Testing team, they were able to meet all the requests of the product owner within the time allotted for the sprint. Testers gave critical feedback to the Developers, helping them to root out bugs in the software and tighten the code. As user stories were updated with the new requirements for wellness vacations, the Developer team modified the expected results of these features, and helped the Testers by creating test cases for each new feature. The Developer team ultimately delivered a working product to the product owner at the end of the sprint.

When the user stories were made available to the team, development of the highest priority stories gave the team the opportunity to focus on developing these first, ensuring that what the product owner deemed most important was accomplished first. This gave the team a time-buffer during the sprint; should the higher priority items take longer, they had the time to complete them as necessary. However, they were able to finish these features as well as the lower priority features within the time limit thanks to the Agile approach to development. With the feedback given to the team mid-sprint, it was even easy to make changes to the product to meet the demands of the users the product owner met with.

When considering the complexity of the Agile method in developing this product, the question may be asked: was it worth it? The answer, in this Scrum Master’s opinion, is a resounding ‘yes’. The Agile method, while very different and much more involved than the simple ‘waterfall’ method, really shined when the product owner came back with changes to the final product based on user feedback. The team was able to make the necessary changes quickly and efficiently, without losing any traction in the sprint. The product backlog was able to be modified with the new features based on the relevant feedback, something that would have had to wait until the final product was delivered had the team been using the waterfall method.

As scrum master, communication is of utmost importance to keep the team running smoothly towards the end-goal of product delivery. Simple email communication kept everyone in touch at all times, while maintaining an information radiator allowed all team members to see overall progress. Daily standups gave everyone a voice, and as scrum master I facilitated those standups to ensure everyone stayed on topic and new issues were discussed as needed at later times after the standup. The standup gave developers a chance to form pairs as well for more efficient programming sessions. Some other tools to use are a centralized board to physically display sprint goals and accomplishments (some people still like the tactile feedback given when physically moving items and removing finished things off the board). For everyone to access, an online tool is utilized so that even remote workers can stay on the same page as the rest of the team. Both of these are updated by me, the Scrum master, as needed.

Just like anything, there are both positive and negative aspects to an agile approach to software development. Unlike a traditional waterfall approach, agile implements and encourages a self-sufficient, self-leading team. This helps give team members ownership over their work, and builds them into a more cohesive unit overall. However, some people are not comfortable in a team where there is no clearly defined leader. The scrum master is a leader of sorts, but the team is designed to work without the scrum master, and therefore they must lead themselves. Generally, this doesn’t create such a hindrance as to completely write-off agile as a useful approach; in fact, once everyone on the team has bought into the approach, it allows them to pivot much more quickly and easily due to how closely they work together and communicate. While the scrum master disseminates information to the team regularly, the team is expected to communicate important information to one another, and this is done more easily in a team that essentially runs itself.

In the end, for the SNHU Travel website build project, an agile approach was the best approach to use, as proven by how quickly the team was able to refactor the project when the product manager requested a change. The Travel site required many different features to be developed, but not all were of highest priority. Being able to break features into stories, have developers work on them through the sprint, and get immediate feedback for what worked and what did not was key to delivering the finished product on time and with the features that were most needed for the market. When a change was to be implemented, developers were able to quickly add in features that previously hadn’t been discussed because they received feedback well-before product delivery day. Had this been done in a classic waterfall fashion, the finished product may have been delivered later than desired, due to having to rework certain features. Giving the team the opportunity to immediately start redevelopment where needed kept the project moving forward, without wasting time delivering a sub-par product.