Andrew Biggs

4/10/2023

CS-250

Sprint Review and Retrospective

All of us in our respective roles have contributed to the progress of this company and the success of this project. Throughout the process, the Product Owner has remained in communication with the stakeholders of the product. Our Product Owner has relayed product specification priorities in the form of user stories in the Product Backlog. This practice allowed the entire team effective communication and to plan our time accordingly. The Product Owner has also been available to the team to clarify any of our questions and concerns regarding product requirements and user specifications.

The Testers on the team have worked closely with the Product Owner to build software tests to ensure the goals summarized in the Product Backlog are met. Each specification is quantified into criteria that the Testers then use to design appropriate tests to make sure every requirement makes it into the release. The Testers work closely with Developers, so everyone understands the specifications of the tests. This enables the team to concurrently develop tests and working software.

In addition to communicating with the Testers during the Sprint, the Developers have communicated with one another to organize themselves as best will benefit the team according to their individual skillsets. Scrum events help facilitate these communications, ensuring everyone is comfortable with both giving and taking recommendations and suggestions about what course of action is best. Problems like poorly designed code, poorly designed tests, and false representation of product specifications are avoided by keeping communication frequent and fresh. This approach ensures everyone on the team recognizes that every one of us is responsible for the quality of the final product- we win together and if we lose, we lose together.

As the Scrum Master, it’s been my duty to facilitate all communications between team members in a way that ensures we are the most agile team possible. Much of this involves allowing the team to self-organize, but there are key moments in which guidance or intervention is necessary. For example, I work to remove any impediments that team members mention to me, and also prevent “experts” from becoming siloed in their work. I ensure that everyone understands the importance of all Scrum events and help people with questions understand how why agile practices are beneficial to the team, individuals on the team, and the company as a whole.

During the Sprint, one of the primary communication tools we use is User Stories, which portray a feature or small set of features of a release in a succinct and objective way. Although any project must have well-defined criteria for completion, Scrum has allowed us to zoom in on one aspect of the project at a time. We learn a lot about our fellow team members by focusing on how to best produce one small increment of “Done” at a time. For example, it’s easy to sit around for weeks debating and delegating who would do what job the best based on preferences, work history, and expertise- but that’s also accomplished while getting hands-on experience together during a Sprint. The process of Sprinting allowed team members who were less confident than their colleagues who were “experts” in some area to both sharpen their skills and lend a hand to expedite the project completion. Then, at the end of every Sprint, we’re able to realize how much we’ve grown both as developers in our own skillsets and as a team made up of members who understand one another on a deeper level than resumés and interviews are able to reach.

The greatest boon the agile-Scrum approach brings to this project is that it makes a change in the project’s direction a minor inconvenience or simple change in direction to the company rather than a dilemma that demands a restructuring of the entire project lifecycle. This is agile-Scrum is a force that pushes the team to develop more modular software. Since each Sprint is only one to a few weeks, we’re planning for the code we’re writing today to be integrated tomorrow into code we haven’t even *thought about* yet. For example, when SNHU Travel wanted to add more options for ways to browse vacation plans on the website, it was easy for the Product Owner to add a few user stories and bump them to the top of the Product Backlog to be worked on in the next Sprint. Although this was a small change in the project’s direction, without agile-Scrum principles and communication practices, the stakeholders could have potentially waited much longer for their change in priorities to be reflected in the release.

Several agile-Scrum organizational tools added exceptional value to the team’s communication and development processes. Our Sprint Planning and Daily Scrum sessions often involve discussions about story points and what precisely constitutes a story point. This made these Scrum events excellent for team building and getting to know one another’s skillsets. Our continual reflection on burn-down charts aided in our discussions by providing us with some of the long-term goal and accomplishment satisfaction that’s achieved in a waterfall process. Jira software made this entire process seamless for not only our Sprint teams, but across the entire company’s development teams. This was especially true in the case of projects with cross-dependency between teams. For example, when our Product Backlog contains items that will be crucial to the implementation of Product Backlog items for another team’s Sprint or release, Jira allows us to easily track both our team’s priorities and plan accordingly.

In the case of the SNHU Travel project, using an agile-Scrum approach was preferred to following a waterfall plan-driven process. This is because during the project, the stakeholders had a shift in priorities that involved switching from a list view of the vacation options to a slideshow view. The stakeholders also added some specifications for features with various priority levels which were also easily converted into user stories and added to the appropriate places in the Product Backlog. These changes were easy to communicate to the development team because we were already checking the Product Backlog for changes and never planning for more than a few weeks in advance. If we’d used a waterfall model to plan this project, we may have had to restart the planning process several times or waited much longer for the stakeholders’ changes to be communicated and implemented.

References

Atlassian. (n.d.). *Jira: Issue & project tracking software*. Atlassian. Retrieved April 13, 2023, from https://www.atlassian.com/software/jira