Project 3: A Sprint Retrospective

Jordan Dayedes

Southern New Hampshire University

CS-250 Software Development Lifecycle

October 13, 2023

Parul Hirpara

This sprint retrospective will lay out the contributions, successes, and learnings from the latest sprint. Everyone was new to agile on the SNHU Travel project. This was a great learning experience for the team and me as a new Scrum Master. I especially needed to be available to coach and help the rest of the team learn what agile is, and how best to adhere to the framework and principles. By laying out expectations of each event early in the process, the team has plenty of opportunity to understand the framework structure.

The product owner spent their time on this sprint working closely with the SNHU Travel clients. With their meetings they gathered information as to what the product expectations needed to be. User stories were then created based off feedback from the client, to be distributed to the development team. While the Product Owner did a great job of providing detailed product stories, there seems to have been some miscommunication, as the stories didn’t line up exactly with what the client needed. (More on that later.) The product stories also needed clarification from the testing team. They couldn’t create solid tests from the information provided in the stories alone, and a subsequent clarification email needed to be sent out.

This all was turned on its head halfway through the sprint, since the client requested large changes to the product. However, both the testing and development teams took in stride. Although new to agile, the only request was from the development team sending a clarification email to the product owner over concerns that the new changes would interrupt the current sprint or be pushed until the next. By keeping with the agile philosophy, the changes were implemented without much interruption to the timeline on the current sprint. The Testers also needed to resubmit new tests that conformed to the new product expectations.

The agile approach to the SDLC is a framework with which a team can tackle a project. The first phase of the project was when the Product Owner met and collaborated with the SNHU Travel client to gather project requirements and build the initial stories themselves. Once the requirements have been laid out, the team can get together early on to discuss and analyze these requirements. This was accomplished from my assignment with which I played the role of the product owner. During this process I listened to a version of a Product Owner having a meeting with clients to discuss product requirements. Then translating those requirements into User Stories to deliver to the development team. Later in the course I also played the role of the developer. Where I took code the rest of the “team” had written and finished my assigned user story by building onto the code base to meet the requirements laid out. The next week in the course I was in the role of a developer writing tests. During which I was given test stories from the “Product Owner”. Since the agile SDLC allows for flexibility for when testing is done, every developer is a tester. Writing tests should be accomplished along side completing assigned user stories.

In week five the client and project manager met again and laid out updated product requirements. Since the agile framework allows for new requirements to be slotted into the existing development pipeline. New changes were able to be added in without the need to adjust timelines for the current sprint. The product owner handled decisions on any user stories that may need to be updated, scrapped, or any finished stories that may need to be brought back and reworked again. Agile also prefers open communication, so any questions that may arise can be asked and addressed quickly. An example of this was during the week 5 assignment playing out this scenario. While playing the role of a developer, I sent a mock email to my Product Owner asking for more details regarding the new requirements changes. Especially any details as to what would need to be addressed during the current sprint.

There were several different types of communication practiced throughout this course. As the product owner I was required to create user stories that communicated product requirements to the development team. As well as test stories that detailed even more specific pass/fail requirements for those same stories. There were also several journal assignments with which I needed to write an email to communicate from one role to another.

Many different aspects of the agile methodology were simulated during my time working through this course. The most notable one for me was utilizing user stories. These felt like miniature tasks that would be manageable by one person over the course of a single day or two. As well as being a key tool in the agile toolbox that’s utilized by nearly every member of the team, from management to the developers themselves. With all other tools building off the user story tasks in some way or another. From time estimation calculating user story points, to keeping track of requirements or maintaining tasks in the event the product changes in a significant during development.

Several pros to agile have been listed already, but arguably the biggest is flexibility. Through scrum events teams can maintain a high tempo by meeting and communicating with each other frequently issues and be found quickly and addressed. Since major decisions are made as late as possible, new requirements can be added mid cycle without derailing a project. However, agile can have difficulty working at scale. When teams can’t collaborate in proximity communication is harder, and so then key agile methodologies have more difficulty. With all of that said, I think a Scrum-agile approach for a project like SNHU Travel would probably be great. Since the team seemed rather small, and the scope of the project wasn’t too extreme. The simulated events also seemed to imply that everyone was working out of the same location, which would help as well.