Madison Lopert

Professor Farley

CS-250-T1138 Software Development Lifecycle

15 October 2022

Sprint Review and Retrospective:

A Scrum team provides different roles that all contribute to the development of a project. The four primary roles in a Scrum team are the product owner, Scrum master, developer, and tester. Throughout the SNHU Travel project, we saw all these roles in action. The product owner is responsible for providing direction to the team, creating and prioritizing the product backlog, and meeting with clients to update them and gather further requirements. In the SNHU Travel project, we saw the product owner Christy, participate in many client and customer-facing events, discussing development options and gathering feedback. The Scrum master is responsible for facilitating Scrum events, providing aid to the team when needed, and ensuring product backlog management. There are multiple Scrum events including sprint planning, daily stand-ups, sprint review and retrospective, and backlog refinement. A Scrum master does not need to be technical; they are present in each of these events to lead the team in contribution. An example of a responsibility that Ron might have as the Scrum master in this project is to facilitate the daily stand-ups. Keeping track of the fifteen-minute time frame and encouraging the team to speak on what they did yesterday, what they will do today, and any roadblocks they might be facing. The development team is focused on completing the stories that they have been assigned. Using the stories as a roadmap, developers are dedicated to completing the story requirements. In this project, we had the opportunity to contribute as developers to fulfill a user story we had on the top five destinations list. We did this by updating the images and descriptions listed on the slideshow list that SNHU Travel was looking to incorporate on their website. In a Scrum team, the tester is responsible for defining acceptance criteria and test cases. In this project, we took on the role of a tester by reviewing the user stories and creating test cases from a user perspective. These test cases include test steps, expected results, and pre-conditions. An example of a test case that I created was how a user travels to the “Destinations for travel” page, where they would view the category filter on that page, how they would interact with it, and what they should see. When a tester notices a defect in any of the solutions that a developer designs, it is their responsibility to make the developer aware so they can solve the problem before it is available to the public.

An Agile software development lifecycle follows these six processes requirement gathering, analysis, designing, coding, testing, and maintenance. The product owner is the primary role responsible for requirement gathering. Christy met with the SNHU Travel client, Amanda, initially to discuss what they are exactly looking to achieve. She also met with the customers to gather some feedback on what they would like out of the product we develop. Using these requirements, she can create and prioritize the backlog. Using the backlog and requirements, the team will participate in Sprint planning to decide what the focus will be on for the first sprint. Once the user stories have been created by the product owner, they will be divided among the development team. The development of user stories was something we had the opportunity to complete during the SNHU Travel project. Utilizing the product backlog, we could designate and create a story ID, size, value statement, and acceptance criteria. The important structure to remember when creating user stories is as a specific type of user, I would like to perform some tasks so that I can achieve something. Once the development for these stories is complete and the test cases for each have been developed by the tester, the testing phase begins. The tester ensures that everything is working as expected from a user perspective. Once cleared by them, the product can go through QA and UAT testing. Any updates that need to be made to the product after it has gone live, fall under the maintenance phase. This could include upgrading the platform, solving defects, or working on new requirements.

The Scrum-agile approach is designed to adapt to change. It is an expected part of the process, so when change does happen, the methodology allows for some flexibility. As a developer, this doesn’t mean the work you have developed will necessarily be scrapped but instead might need to be updated. We saw an example of this while working on the SNHU Travel project. The product owner met with SNHU’s Travel Management team that found an industry report saying that detox/wellness vacations are going to be the next big travel sector. With this knowledge, as a developer, I could come up with some ideas and locations I felt fitting to include on the slideshow. These changes included a background color, destinations that offer detox programs, and a description of the leading top destination. While changes can be made, it is important to discuss them with the client beforehand. If any additional information is required during or after development, like in our case, an email or meeting can be established to ask questions about what the client is looking for, or if a change that has been made is fitting. If a change like this happens the product owner will update and/or create the user stories and communicate their changes with the team. The testers will update the test cases to reflect those changes.

During the team collaboration assignment, I operated in the tester role. An example of my communication in this discussion is, “Matt, I will make sure that I create a good baseline for the test cases before the PI planning meeting. Using the user stories in the product backlog, I will be able to create the test cases for the first sprint.” I sent this to the group after reading through what my teammates had already communicated. Matt, the product owner, had let the team know he would meet with the developers and testers during the PI planning meeting and ensure that tests are put into place. My example shows my response by explaining my next steps and deliverables for when the time to attend the PI planning meeting happens. Another example of my communication is shown through my direct response to Jordan. “Per information, I have received from the team I see we have decided on the Jira software as our radiator. This is great news! I now know there is a reliable location to view stories, track their progress, and develop test cases for. Please feel free to reach out if you notice anything you would like to add or remove from my test cases, and I'll make sure to meet with you if I need more information about the configurations.” This sample shows my acknowledgment of the software my team has decided to utilize, as well as forms that direct line of communication that is essential to a tester and developer relationship. Letting Jordan know that I will reach out to them if I need more information about configurations and that they can reach out to me if they ever notice a way to improve my test cases establishes an understanding of what our interactions might look like.

A couple of organizational tools that a team could utilize in a Scrum-agile environment are JIRA and Microsoft Azure Boards. The commonality between these two is the storyboards they offer. Storyboards are essential to the Scrum-agile process and help immensely in showing what is currently being developed in each sprint. Each user story holds its place on the board and is often separated based on its current stage in development. For example, work in progress, assigned, development complete, etc. These storyboards are designed and managed by the Scrum team to ensure that work is properly reflected. With the many events that Scrum holds the storyboard can be utilized differently. The Scrum events are as follows, sprint planning, daily scrum, sprint review, and sprint retrospective. Daily scrums are going to utilize the storyboard more directly than the others. The Scrum master will have the organizational tool up for the team to see and might pull up specific stories to receive updates from the development team and testers. Depending on the feedback, the Scrum master might update the story but ideally, the development team should make and communicate those changes to allow more time for discussion during stand-up. The storyboard reflects the current Sprint and the decisions made for which stories are available will be decided in the sprint planning sessions. The storyboard might be utilized in a sprint review session to show the client story details if needed. A sprint retrospective won’t need to have the organizational tool present, since this is more of a memory and individual experience-based event. Always having the story details available and effectively updated by the team will ensure that everyone on the team knows what to do when to do it, and what they should prioritize.

The Scrum-agile process is a useful tool to develop a product. With any methodology, there are pros and cons to how it operates. In the SNHU Travel project, agile helped in communication through an information radiator that listed out the story details. The Scrum roles made it simple for the team to understand their responsibilities throughout the development process. Being an agile process, Scrum was built for change. When SNHU Travel management wanted the product to focus on detox/wellness vacations, that was possible to integrate within the designated time frame we had allotted for development. Some cons that might come from agile are limited documentation, poor resource planning, and extended due dates. If someone is not designated to develop documentation specifically it often gets overlooked. Each story can be estimated through story points. Depending on the experience of the team, these story points might be improperly mapped, and the team might take on too little or too much in one sprint. While agile was created to adapt to change, new requirements and alterations to the product might take longer to develop than expected. This could result in the extension of a sprint or deliverable. While there are some cons, the Scrum-agile approach was the best choice for the SNHU Travel team. The waterfall approach is a rigid process that doesn’t allow for any repetitions of a phase. Meaning when requirement gathering is completed it can’t be reintroduced. If a change is wanted from the client, it just isn’t as easy to incorporate it in a waterfall structure as it is in an agile one. SNHU Travel’s need to transition the product to focus on detox/wellness in the middle of development is just one example of something that could be done in agile but not in waterfall.

References

SDLC - Agile Model. Tutorials Point. (n.d.). Retrieved October 16, 2022, from fffffffffhttps://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm

SDLC - Waterfall Model. Tutorials Point. (n.d.). Retrieved October 16, 2022, from fffffffffhttps://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm

SNHU Media. CS250-module five: Product owner and Scrum-Agile Team. (n.d.). Retrieved fffffffffOctober 16, 2022, from http://snhu-ffffffffffmedia.snhu.edu/files/course\_repository/undergraduate/cs/cs250/storyline/mod5/story\_htffffffml5.html

SNHU Media. CS250-module three: Product owner and User Focus Group. (n.d.). Retrieved ffffffffffOctober 16, 2022, from http://snhu-ffffffffffmedia.snhu.edu/files/course\_repository/undergraduate/cs/cs250/storyline/mod3/story\_htffffffml5.html

SNHU Media. CS250-module two: Initial client meeting. (n.d.). Retrieved October 16, 2022, fffffffffffrom http://snhu-ffffffffffmedia.snhu.edu/files/course\_repository/undergraduate/cs/cs250/storyline/mod2/story\_htffffffml5.html

Success rates rise - project management institute. (n.d.). Retrieved October 16, 2022, from ffffffffffhttps://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-ffffffffffleadership/pulse/pulse-of-the-profession-2017.pdf?sc\_lang\_temp=en