Roderick Flowers

CS 250 Module 7-1

June 23, 2024

Throughout this course, I have taken on the role of various members of a Scrum team. For these assignments, I have been assigned to the SNHU Travel team as they transition from the waterfall approach to the more widely accepted Agile method in preparation for the development of a new travel application. The roles assigned in an Agile team are the Scrum Master, Product Owner, and the Development Team, the latter of which consisted of developers and testers. This paper will serve as a retrospective on the lessons learned throughout this course.

The Product Owner understands the customer and business requirements, then creates and manages the product backlog based on those requirements. Since agile teams are, by design, flexible and responsive, it is the responsibility of the product owner to ensure that they are delivering the most value. The product owner also balances the needs of other stakeholders in the organization. So, the product owner’s most important job is to take all these inputs and prioritize the work.

The Scrum Master is the role responsible for gluing everything together and ensuring that scrum is being done well. That means they help the product owner define value, the development team deliver the value, and the scrum team to get better. They serve the product owner by helping them better understand and communicate value, to manage the backlog, help them plan the work with the team and break down that work to deliver the most effective learning. The Scrum Master must focus on transparency, empiricism, self-organization, and values. The scrum master ensures that every opportunity to improve is made transparent to the scrum team and the retrospective has a clear set of outcomes that can be executed.

The development team are the developers, testers, writers, designers, and anyone else who is doing the work regarding the project. The development team should be able to self-organize so they can make decisions to get work done. The development team must deliver the work through the sprints the scrum team has. They, like the scrum master, must also ensure transparency during the sprint the meet daily at the daily standup. This allows them to discuss progress and any roadblocks that may have occurred during their work. This also allows the team to highlight successes and good habits made along the way.

The Scrum-Agile methodology in software development lifecycle separates crucial functionality within projects. Proper execution is essential for navigating the complexity of software planning. Breaking down intricate tasks into smaller increments is pivotal for achieving successful deployment of the project. For the travel project, I got all the requirements directly from end-users and made user stories with them. These stories were concise, yet they sufficiently described the functionality required by these specifications. A typical User Story follows a standard format: it identifies the user, specifies what task they need to accomplish, and clarifies the purpose or value of that functionality. This structured approach ensures clarity and alignment throughout the development process.

The agile methodology is designed to be flexible and responsive to changes during the software development process. It is expected that things will change before a final product is delivered to the stakeholders. During the SNHU Travel project, a change was made to focus more on detox/wellness travel. I was then tasked with revising code that was already developed in order to support the new requirements.

During the project, I was instructed to make an email regarding the changes to the direction of the product. In the email, I requested that an in-person meeting be held to discuss the changes and to ensure that everyone was on the same page. Here is an example of the communication technique I used for my assignments:

*To: Rod (Product Owner) & Zaylend (App Tester)*

*Subject: Request for Additional Information*

*Hello Rod and Zaylend:*

*I am a developer for SNHU Travel software. I am currently working on implementing the changes you requested to meet the new requirements of the software. I have a few requests to make to ensure that I meet the exact requirements you all have added.*

*To start, would you provide me with more information on how the software’s layout and design should be to meet the client’s needs? Any additional information you have to provide the client with the highest quality product possible would be very helpful as well.*

*Secondly, would you be able to provide some updated test cases that meet the new requirements of the project? These will help ensure the functionality of the product.*

*I would like to set up a meeting with the both of you so that we can discuss these matters clearly and make sure that we are all on the same page. I feel as though the emails and messages are taking too long to respond to and some things are getting lost in translation.*

*Thank you,*

*Roderick Flowers*

I believe that my email was concise and got to the point I was trying to make. My biggest concern for this email was to set up an in-person meeting as I feel that going back and forth with emails would be a waste of time and could lead to potential roadblocks as intentions are easily misunderstood through text.

The tools that helped me transition this team from a waterfall method to an agile method were JIRA and Azure DevOps. Azure DevOps allowed the team to facilitate the workload via the introduction of product backlogs, user stories, and sprints. I think these were crucial in the transition because they gave me a clear workload that allowed for the easy implementation of the changes that happened later in the course. I used JIRA to manage the individual tasks that needed the team’s attention. Using these tools also aided in the team’s transparency as the daily sprints would allow for everyone to discuss their roadblocks and successes as well as provide a platform for each member to see if another member has completed their work.

I think transitioning the SNHU Travel project from waterfall to agile methodology proved to be highly beneficial to the product. It allowed the team to have greater transparency and far more flexibility than the waterfall method would have allowed. The key point for that is, again, the sudden change of direction to detox/wellness traveling. With the waterfall method, the team would have pushed a product before finding out about this change and would have to scrap the product and basically restart while trying to stay within the deadline. The agile method allowed for this change to happen mid-term and the team was able to adjust to fit these new requirements immediately with barely any time lost. The issue I could see with the agile method is because it allows for such flexibility, the scope of the project can be easily blown over budget. It could also have so much added during development that it loses the initial idea of what it was supposed to be. Overall, though, I believe the agile method is the best way for a team to develop software.