CS-250 Software Development Lifecycle

Brandon Thaller

In this course I took on multiple roles that make up an Agile team. Some of these roles include Scrum Master, Product Owner, Developer, Tester. Taking on the role as Scrum Master it is my responsibility to coordinate with the Product Owner to ensure the customer is getting the product they may be envisioning. As the Scrum Master it is my responsibility to host the daily scrum to set schedules and the timeframe for each section of the project. These daily meetings keep the team in constant feedback to understand where everyone is at and what the next steps moving forward may be. Another role of the Scrum Master is to coordinate with the Product Owner to ensure we are prioritizing what the client wants.

As the Product Owner we are the face of the team, we are in constant contact with the client. As the Product Owner you need to understand what the client wants and what the capabilities of the team are, this helps set initial timelines and set expectations. The Product Owner will work with a test to compile user stories for the potential features and how they work.

Another role is the Developer, as the Developer you are to use the test cases to compile the features. Testers and Developers work together to refine the user stories and the features. The Developer is responsible for the current development of the features and while in the daily scrum they can speak up to where they are at, and where they may need assistance. When the Developers complete a feature, they get tested by the Tester to determine if the feature can be refined and improved or if it meets the requirements of the client.

Teams in agile are often small, inciting productivity not having many people too involved in a project, but also requiring the team to work together and help work through shortcomings. Every member of the team is meaningful towards the end of the project. While the Waterfall Method works in smaller projects, during larger projects it is less effective. The Waterfall Method works towards project completion with little testing, causing problems when unexpected issues or bugs arise during the testing process before rolling out. This sometimes gives little time to correct issues before release. Opposed to the Agile Method testing is done in phases and user stories get refined on how the feature works during testing. This creates a stable product launch and allows the team to focus on discovered bugs after launch.

The Agile Method giving time for testing makes it easy to change the project's approach, due to the testing being done in phases. The Waterfall Method leaves no time, and a change in the project could cause considerable time waste by scrapping all the work. Giving Agile the edge to decent turnaround times for a project.

In Agile because user stories are tested during phases and being easily refined, these things can be communicated with the Product Owner and then portrayed to the client to continue to refine the features. Having this open communication between the team members makes this extremely effective and allows us to adjust times frames and set new priorities.

The Scrum Master needs to be aware of where the project is and what the main priorities are, this tends to happen on a whiteboard, or sometimes on a web-based service. This approach to management is heavily dependent on members being active and speaking up if something is taking longer than expected to the Scrum Master could assign help.

Some factors to consider when choosing a Waterfall approach or an Agile approach is the size of the project, smaller projects tend to work well in the Waterfall method due to not needing to rework an entire project due to a singular feature. Agile works the best on larger projects because the features can change and needing to scrap all the work and start anew can cause massive time defferments for the estimated deliverable time.

I think a Scrum-agile approach worked fine for this application but could've been done with the Waterfall method as well. I think the SNHU Travel project falls in between the large and small project size. Some pros are allowing time to refine what the client wants and having a fast turnaround time. Some cons may be that Agile can sometimes cause project to run longer due to the meetings and constant conversation.