Sprint Review and Retrospective

The Agile Methodology (Agile) is a product development process associated with software development. Agile’s focus is to continuously improve on the product throughout the development lifecycle. Agile uses smaller teams (Scrum Teams) to bring project to completion. Each team is comprised of four specific roles. These roles are Product Owner, Scrum Master, Developer, and Tester. Each role is equally necessary for a successful Agile project. The teams work towards smaller goals throughout sprints, which are typically one to two weeks’ worth of work.

The Product Owner (PO) is relatable to the typical project manager. The PO is responsible for producing stories and prioritizing the Product Backlog, which is a log of all the tasks for a given sprint. The PO keeps the team on track and can sometimes even help with tasks if they have the bandwidth. The PO will also schedule meetings with clients and/or stakeholders. If some tasks cannot get completed within a sprint or if some unforeseen obstacles get in the way, the PO will then need to groom the product backlog. Grooming the backlog is prioritizing and deprioritizing tasks, sometimes even removing or adding tasks. For example, while working with SNHU Travel instead of having a list of items on their webpage the client wanted a slideshow. The PO had to groom the backlog to accommodate the new client request.

The Scrum Master (SM) is, in some ways, a little like the PO, but they work more closely with the scrum team members. I see Scrum Masters as team leaders. The SM will analyze the stories and backlog from the PO and work with the team to determine timelines. SMs will hold daily standup meetings, which are short meetings that last no longer than fifteen minutes. Standups are for the team members to give their status in the form of: What did you accomplish yesterday? What will you accomplish today? Are there any obstacles in your way? There are also team meetings that can be held once a week to go over tasks in more detail. An example would be when SNHU Travel wanted to change things. After the PO updated the backlog, the SM had to inform the team and help create new timelines.

The Developer is responsible for a deliverable product. They also understand the definition of done (DoD), which is when all the acceptance criteria are met. The role of developer can vary from company to company and/or project to project. Some teams/companies may have stricter or more lenient DoD. The developer is usually doing the coding when it comes to software development. An example from the SNHU Travel project was that the developer built the new list of top destinations. The developer also had to rework their code to comply with the client’s updated requests for the project.

The Tester writes tests to assess the features that the developer produced. Sometimes the tester and developer are the same role. For example, some companies want developers to do what is called Test Driven Development (TDD). TDD is when the developer will write the test before he/she writes the code for the feature. In our case, working with SNHU Travel, the tester and developer are two separate roles. The tester will look at the stories and produce test cases for each one. The developer will just write the code for the feature.

The Scrum-agile approach helped with the completion of each user story. Having short sprints is beneficial because you can catch errors and update accordingly much sooner than the Waterfall approach, where these errors would not be handled until the end of the project. An example of this while working with SNHU Travel is how the team produced detailed spreadsheets describing each user story, the acceptance criteria, the priority level, and the size of the task. Each role knew exactly what to do and how to do it. The detailed user stories made it extremely easy to hand off to other team members if needed. The sprint approach was extremely helpful when SNHU Travel wanted to make changes midway through the project. During a Waterfall Software Development Lifecycle (SDLC) this change would not have been able to be implemented. With agile this was a bit of a speed bump, but it was easy to fix. The PO had to groom the backlog, the SM had to make sure the team knew what needed to be done, the developer had to do some more coding, and the tester had to update their tests accordingly.

The Agile approach also encourages collaboration within the team. Without good collaboration, Agile would not work as well. While working as a tester and developer for SNHU Travel I had to reach out to other team members to get some clarity on certain things. The following are two emails I sent out to the Product Owner:

As the Tester:

To: Product Owner

Subject: User Story Clarifications

Hi Product Owner,

As I was going through the user stories for our current sprint a couple of questions arose. Can you help give me a little more clarity on the following stories?

User Story Two

* Should the “Hot Deals” page be ordered by preferred destination for the profile, or should it be ordered by the price (cheapest to most expensive)?

General Questions

* Are the stories prioritized by their ID, size, or something else unknown to me?

Thank You,

Tony

As the Developer:

Hey there Product Owner,

I just have a couple request that I am wondering if you could help me out with quick.

1. Do we need to personalize these detox/wellness vacations depending on the profile?
2. Should these destinations be worldwide, or should they be limited to a specific region?
3. Should I finish up what I was working on, or stop short and come back after the new changes are made?

Let me know what you think!

Thank You,

Tony

There are many tools a company can use to increase the effectiveness of the Agile work methodology. These tools can range from GitHub, Excel, Eclipse IDE, to JIRA. While working on the SNHU Travel project we used the Eclipse IDE and Excel. Excel was used to keep track of the product backlog, write user stories, and keep track of the test cases. Excel is always a great tool to keep track of any project’s progress and is highly effective for Agile. The Eclipse IDE is one of the tools I am not too familiar with. I personally do not like it, but it does make coding with Java much easier, which was effective for this specific Agile project. I have used JIRA and GitHub for other projects, and they are both remarkably effective. GitHub or GitLab are a must-have for most software development projects. These tools are incredibly effective for collaborating with other team members and other teams. JIRA is effective because it is a suitable place to audit test results. I have used JIRA with an Agile project to keep track of nightly test and to write tasks for Kanban boards. This tool is also highly effective for collaborating with others.

Overall, I do think that the Scrum-Agile approach worked well with the SNHU Travel project. I may be biased, but the Agile approach is always a good fit when working on software development. The pros for it being:

* The sprints made completing the smaller feature requests more manageable
* The change of direction was managed elegantly
* Collaboration was encouraged throughout the project which helped with the time of completion

The cons:

* The change of direction
* Limited documentation