Danny Olf

02/20/2021

CS-250

Sprint Review and Retrospective

The SNHU Travel project was designed to give the students a first hand feel for the Scrum-agile methodology. To accomplish this, the assignment has the student take on the different roles associated with Scrum. When I was assigned to be the product owner, my job was gather information form the client and some of the users to collect the user stories. From there my job was to create a product backlog. This was crucial to understand exactly what the user was looking for in the features being requested. Collecting user stories let the me (as the Product Owner) communicate what the users were trying to say extremely effectively. By gathering the user stories straight from the source, so to speak, I was able to pass that information to the rest of the team so they could keep the end goal in mind. One the main ideas behind Scrum to stay focused on the problem. This may seem counterintuitive at first. However, by focusing on the problem the development team can find the best possible solution. If the team gets too focused on the solution, they could waste time, and therefor money, trying to force that solution when it may be more effective to find a different one. People tend to get attached to their ideas, and this can end up costing the company money or slowing down the project. In the worst case, it can bring development to a complete halt. As the Scrum Master, my job was to take the product backlog and organize the list into small bite sized pieces known as Sprints. This way I was able to keep the team moving and at the end of each Sprint, the team can asses how they did and how they can improve for the next iteration. The Scrum Master also acts the middleman (for lack of a better term) between the development team and the Product Owner, and is in charge of setting up meetings and keeping the team on track. As a developer, my goal was to decide how many user stories could be finished in a given Sprint and to give the Scrum Master feedback regarding setbacks and things we could do better in the future. When I was assigned the role of a Tester, my job was to look the features for the Sprint and figure out a definition of done for each feature. This was done by defining criteria for each feature (gathered from the user stories) so that we could test them efficiently. If they did not meet the criteria, they test was failed and the feature could not be considered done. In doing this, the Product Owner could convey to the client more accurately what was done and what was not. It takes all these different team members to make the process work as well as it does. The Scrum-agile approach does a great job at delivering value early and consistently.

The Scrum-agile approach helped with the user stories, first by introducing the concept of user stories. A user story is where the client or user describes something they want in a way that is simple and concise. Instead of a multi-page document that is extremely technical, a user story can be described by saying, “As a [type of user], I want [a goal] so that I can [a reason].” Usually this is one or two sentences. As stated previously, this reduces the tendency for developers to get too focused on the solution. Staying focused on the problem is key. For instance, on elf the user stories in the SNHU Travel project was give the user the ability to see the top five travel destinations recommend for them. Using the format above, it would look like this: “As an End User, I want to be able to see the top five travel destinations so I that I can see the most popular locations for travel to educate myself on the best places to go.” This format keeps the team focused on the goal that user is trying achieve instead on meeting the requirements of a technical document.

The main advantage that the Scrum-agile approach has over the more traditional Waterfall approach is that it (as the name suggests) is flexible. Projects tend to change over time, and as such the team working on it need to be able to adapt. This is where Waterfall fail fails and Scrum excels. Scrum is designed to be adaptable. To illustrate this, one of the SNHU Travel assignments was to adapt the project to a new set of requirements. Since the product backlog had been broken down into short sprints, this was relatively easy to accomplish. If the Waterfall approach had been used, a ton of work would have been wasted, and there would have been a chunk of time solely dedicated to redoing the plan to accommodate the new set of requirements. However, since the Scrum-agile approach had been taken, all that was need was to adjust the plan for that sprint. Even if the feature that the team was working was scrapped completely, it would have only set the project back a week or so at most.

Another crucial aspect of the Scrum-agile approach is open and transparent communication. As part of the SNHU Travel project, I was tasked to write emails to deferent members of the team. Also, one of the discussion assignments was to act as a team that was converting to Scrum from Waterfall. In the SNHU Travel assignment I was able to communicate via email by keeping it short and to the point. For example, in the part of the Travel assent where I was a developer I wrote an email to the Product Owner. In this email I made a list of questions regarding the user stories. Instead of writing a long email asking technical questions, I asked simple, straightforward questions. The goal, as previously stated, was to keep my focus on the problem I was trying to solve. I asked questions such as, “[W]hat exactly is detox/wellness in terms of travel packages?” I could have asked something more technical, such as, “What criteria needs to be met to classify a travel destination as detox or wellness oriented?” By keeping it simple, I was able to get to the point of the question and not waste the Product Owner’s time. Furthermore, by keeping my email short and simple, I was able to encourage open communication with the Product Owner. In the discussion, I was able to do the sam by fallowing that example. I started the discussion by introducing myself, my role and my goals. Thought the post, I reiterated how important transparency was to me, and encouraged open and honest feedback regarding my plans to adopt the Scrum approach.

Organizational tools are a great way to help make Scrum even more efficient. Today, there are a plethora of tools to choose from. Scrum events can be tricky to incorporate. Luckily, many of the tools out there are designed to help. The different Scrum events (Sprint planning, Daily Scrum Meeting, Sprint Review and Retrospective) are at what up Scrum. Each tool has its pros and cons, so it important to consider exactly what is needed when considering which tool to use. I am going to focus on three of the tools I have been introduced to: Jira, Microsoft Azure Boards, and Rational Team Concert. Jira has an impressive list of features that are focused on creating and sharing different boards. They seem to use the Scrum-Kanban approach. Jira also has some features designed to help with cost and time estimations, which is very useful. Azure has many of the same features and is also focused on creating and sharing boards. In my opinion, Azure is the best due to how easy and user friendly it is to use. After all, a tool is only as good as its ability to be used. Last on the list is Rational Team Concert. Unlike the other two, this tool is designed to help share information among the team. This is a really nice thing to focus on because in many instances the team could be scattered across the globe. As I have stated before, communication is a crucial aspect of Scrum, and that is the thing this tool tries to focus on. However, it is lacking, in my opinion, in the ease of use and rich list of features that the other two have. When planing a Sprint, for example, it is important to be able to list things easily, be able to adjust them as the requirements change, and finally share them with the rest of the team. Azure Boards does this really well. During the Daily Scrum it is important that the whole team is involved. Yes, Rational Team Concert does this part well, but so do other tools such as Skype and Zoom. The Review and Retrospectives are similar in that the ability to collect and organize input from the team should be the focus. This is where Azure separates itself from the pack. A board can be made to address all of the issues, and then organized, stored and shared really easily. It is intuitive interface that sets Azure Boards apart.

All in all, the use of the Scrum-agile approach made the SNHU Travel project as painless as possible. It took a big project and broke it down into bite size chunks, set the flow so that it was sustainable, and allowed for changes to be made during project development. It is hard to say whether or not the project would have been successful if the waterfall approach had been used. However, what I can say with complete confidence is that the use of the Scrum methods made the whole process easy to digest and to understand. It also removed the potential roadblocks that tend to come from things such as lack of communication. Overall, I am glad to have been introduced to Scrum. I understand why it is as popular as it is, and only seems to be gaining popularity in the software development field.