Jon Marvin

6/18/2024

CS 230

Agile Sprint Review and Retrospective

SNHU

**Applying Roles**

After weeks of assuming the roles of each Scrum Team member, I have concluded that each role is essential for the success of the project. Every role supports another in many ways, and losing any of them would be remarkably detrimental to the scrum team.

**Scrum Master**

Assuming the role of the Scrum Master, I found that the best strategy was to hold short, effective meetings in the morning and give each team member their chance to speak. Everyone would cover their recent accomplishments, to-do list, and anything that may be holding them back. Sprint planning takes the same type of shape as the daily scrum, but in a longer form meeting to ensure all bases are covered for the team. We want to make sure no stone is left unturned, and no task falls behind.

**Product Owner**

The Product Owner was an interesting role for me to assume. I primarily work on the back end in my career, and communicating with clients and stakeholders is not something I normally do. Taking feedback from users and testers is something I do regularly, so that bit was familiar, but on a different scale. Product Owners need to be social and communicative with everyone they are involved with. This includes stakeholders, members of the team, and users. Taking user stories to help with development, specifically from those using the product, is the most valuable trait of the Product Owner, in my opinion. To be an effective leader, you must listen to your people and act appropriately on their behalf.

**Development Team**

The development team consists of the actual, hands-on developers and the testers. Developers create the actual code in the product, while squashing bugs and making updates with the help of testers and user stories. To be an effective developer, criticism needs to be taken as help rather than offensively. The team is working together toward a common goal, and getting upset over criticism is a sure way to sink the ship. Testers do just that. They test the product and report any bugs they find. The tester can also use user stories to dive deeper into something that may be an issue or something that could be implemented to make the product stronger or more attractive to many users. Testing the product is essential, especially when still developing, to avoid any roadblocks when the product is shipped. Launching a new project full of issues is not something a scrum team should be doing.

**Completing User Stories**

User Stories can be used in a Scrum-Agile approach to tell the developers what the users feel is missing from a product. When SNHU Travel asked users for feedback, they were given a few strong ideas that potentially could have been implemented in development but had not been addressed previously. Though a user story does not represent all users, it more than likely represents the majority, especially if the same idea is heard in multiple interviews. Using User Stories in the Software Development Life Cycle is a key to success in that the missing functionality can be pointed out in a product. A missing search bar, profile customization, and suggested travel plans were all brought up by users for SNHU Travel, and those are things that any user would want to see in such a heavily used app.

**Handling Interruptions**

Interruptions can be game changing. There are many types of interruptions from a developer becoming sick, to the stakeholders turning the entire project around and trying to come up with a new way to create the product. Agile is flexible. If something changes, the team can change with it to deliver the product that the stakeholders want without losing too much time. Uncertainty is part of the Agile process with so many potential roadblocks and U-turns that could appear at a moment’s notice. I have seen this happen in my professional life multiple times. In my role as a data analyst, I am tasked with creating the reports that make their way back to the client. We have had innumerable view changes, metric updates, and dataset swaps to give the client the most granular look at the data they are asking for. They supply the work, so we must give them exactly what they ask for and more.

**Communication**

Communication in any team is one of the most important aspects of understanding a project. If open communication is not established between team members, no one will be on the same page, likely leading to major issues in the project. The sample email that I wrote for the Tester Journal explained that everything looked good in two of the test cases, but one of them was unclear. Simple communication like that can lead to better results and a stronger team. It is important to ask questions if not all the information is supplied, or if something seems confusing or incorrect. I have had to reach out to my superiors at work many times because data that we should have had was missing or incomplete. I have gotten more proactive with it recently; rather than asking what to do, I inform them of the mistake and reach out to the supplier myself so as not to interrupt whatever they may be working on at the time.

**Organizational Tools**

Organizational tools can be used for any number of things. Depending on the application, something as simple as Microsoft Excel can be an excellent tool for a wide variety of projects. Some tools that my team agreed to use for the SNHU Travel project were a Scrum or Kanban board to keep the backlog in order, user story mapping to create a stronger application for our users, and various meetings to keep each other updated on what we have completed. Meetings may not seem like something that can fit into the category of “tools”, but I strongly believe that meetings, calls, and conversations that help others understand something can be considered tools just as much as software and hardware. One other tool that I have experience with is JIRA. The ticketing system has helped my company stay up to date on any items that may need to be fixed or applied, and every team can use it to request something from another team.

**Evaluation**

I found the Scrum-Agile approach familiar when this course started. I have been using it in my career without knowing it, and getting a full understanding of how it works has made so much sense across the board. When applied to the SNHU Travel project, the Scrum-Agile approach is a clear winner. The team understood what was needed to be completed, the product owner had meetings with users to find what may be missing from the app, the Scrum Master helped everyone get together and work through any issues that they may be facing, and the development team implemented any changes that were made from the stakeholders and suggestions from the users. There can be drawbacks to this approach as well, such as potential delays if too many changes are made, going over budget due to the uncertainty of the project at the beginning, and the stakeholders changing lanes at any given point. Changing minds is something that occurs regularly, but when applied to something that has a set due date and budget can lead to disaster.

I feel the Scrum-Agile approach was the best approach for the SNHU Travel project. Hearing from the users alone led to a major improvement in the quality of the product, and I am certain they would appreciate the developers listening to them. Agile also gave the team the flexibility to test on the fly rather than waiting and backtracking to fix bugs and implement suggestions from users and stakeholders, giving the project a more lifelike feeling. The amount of work that goes into a project of this size is tremendous, and teams that can accomplish these goals, given the timelines and updates, are impressive. Had the team used the Waterfall model, they may not have gotten everything implemented that they wanted to.