Sprint Review and Retrospective

Keri E. Ludemann

Abstract

For this paper I’ll be looking back on my past work I’ve done for the SNHU Travel project. I was given the chance to step into each team member’s role which allowed me to experience how each member contributes to the creation of a quality project. Through this I was given glimpses of how team members communicate with each other outside of the daily Scrum Events and meetings. I will be going over the benefits of using an agile approach for this project compared to a waterfall approach on top of explaining the ways teams apply agile principles or tools to create their projects.

Sprint Review and Retrospective

During the SNHU Travel I was given the chance to take on the roles of Product Owner, Developer, Tester and Scrum Master. Being able to take on each of the different roles gave insight into how necessary each member is, how they all cooperate to create a product and how they communicate with each other when more information is necessary for advancement.

The first role I took on was of the Scrum Master. During this role I got to see an example of a Scrum meeting, or Daily Scrum Event, and analyze how it’s beneficial to the team. The Scrum Master has a hand in backlog grooming, sprint planning, the daily Scrum Events, and the sprint retrospective. Backlog grooming, sprint planning and the Daily Scrum events require there to be collaboration and accountability amongst team members. As a Scrum Master it is your job to facilitate these meetings and events to keep the team going by making sure they have nothing impeding their progress. They typically start the days by asking what all the team has achieved, what they are working on next and what obstacles they are facing. If the team mentions things, such as there needs to be less glare on their screens, the Scrum Master will take it upon themselves to fix this problem for them.

The second role was of the Product Owner. While in this role I was given the chance to create user stories which are there to give the developers a chance to see what the client is asking for and give them the opportunity to decide on the technical requirements. The Product Owner helps to prioritize the work which in turn gives direction to the development team. They serve as the middleman between the client and the team, so they are also the ones that inform the team of any necessary changes that need to be made per the request of the client. A way that they provide support to the team is through user stories since it provides information on what exactly the client is wanting along with steps on how the client is expecting it to look and function. Here is an example of a user story I was able to create.

|  |  |
| --- | --- |
| User Story Number: | 2 |
| User Story Name: | Price Ranges |
| User Story Size: | Large |
| User Story Statement: | As an end user I want to be able to choose the prices ranges of vacations when searching so that I can see what vacation packages are available within me price range. |
| Acceptance Criteria: | * Ability to add as a filter for searching * Ability to set a min and max price * Ability to sort them in order from high to low or low to high * Only show vacations within price limit * Show full price of package for the trip instead of just per day/night |

Within the example you can see the number, name, size, a small statement on what is expected and the criteria for how it should operate. When creating the user stories for the developers it is important to list their size or how important the job is. Thanks to this it is easier for the development team to prioritize the work and ensure what needs to be done first. Since the Product Owner also puts down a statement about the item and the criteria it informs the developers on what exactly they need to code for the item to turn out the way the client is wanting.

The next role I played was one of the Tester. While in this role I realized that the Tester works closely with more than just the development team. They also collaborate with the Product Owner when creating the test cases based on the user stories. Through out this there may be times where the Tester needs more information for a test case so they will get ahold of the Product Owner. Here is an example of an email I created as a Tester to the Product Owner.

To: Tester

Subject: User Story Clarifications

Dear Product Owner,

After taking a look at the user stories provided, I would like to make sure that we have everything we need in order to make them happen.

**User Story #1**

* Would the filter/ sort bar come up after they have already started a search? Where on the screen would it be?
* When interacting with it how will it come up? Would it be a bar that comes in from the side? Or maybe take up the whole screen?
* What all will be listed in the filter section? What kind of filters are we going to be adding?

**User Story #2**

* Where would the recommended tab be?
* Will the user be taken to another open tab, or will it be the same tab opening a page?
* How will it show the recommendations? Slide show or scroll down?

**User Story #6**

* Will there only be the listed tabs for the drop down list?
* What other profile settings might the user be able to change?
* Will the added preferences affect the advertised vacations on the home page or not?

Thanks,

Keri

As mentioned in the email, this was sent to get clarification on the acceptance criteria for each user story. As the Tester I needed to know exactly what the client or end user needed to see and how it operated. Sometimes you will need to get in touch with another team member outside of the daily meetings or Scrum Events and this sample email shows that collaboration between team members continues even after those.

The Developer was the last role I played during this project, and it helped to show how all the teamwork came together. Each developer may choose to work on an item by themselves or with someone else when they need help on something. The team members will teach each other while also taking part as a Tester and sometimes the Tester may even work closely with a developer during the coding of a program. They get information for the product from the Product Owner who will also inform the team of any changes that come up during the process of creating the project. Here is a sample email I made while playing the role of developer when a change was mentioned.

Hello <Product Owner Name> and <Tester Name>,

I’m emailing you today because I need some extra information to make sure I’m going about this new plan the right way. First, I need to know the vacation packages I’m to be adding and what details about them they want shown. Next, I need to know exactly how this is supposed to look for the end user. Am I to make it look like a slide show or am I going to make it a scroll down web page?

Thanks,

Keri

As a Developer, being told that changes need to be made to something you have been working on can be stressful. When communicating with the Product Owner and Tester about the changes and what all you may need to rework or completely scrap it makes the process much easier. The email was asking for clarification on the change that had been brought up which may not have been mentioned during the meeting that was called when the client requested it. Changes can pop up during any project and it’s important to handle them in a timely fashion to ensure there aren’t any delays. Communicating and collaborating with team members means it can be addressed quickly while also making sure it’s done right.

Utilizing a Scum-agile approach helped to role with the punches when it came to changes or when more information was added about certain items for the SNHU Travel project. The main principles that I utilized were about the team and team work, working with the customer, and delivering working software over doing in-depth paperwork. The Daily Scrum embodies the principle of teamwork and face-to-face communication through giving everyone a short period during the day to go over the day before, plans for today and the challenges they’re facing. By also utilizing an estimation technique, that works for the team, it is easier to divide projects into manageable chunks or items making it easier for the team to divide the work so that no one has nothing, and one isn’t overburdened.

Thanks to the adaptability that comes with a Scum-agile approach it made this project easier to do. Since the process doesn’t follow a strict plan it allows the customer to request changes and allows the team to roll with the punches when they come. I didn’t come across any cons when working on this project however I feel that not having more information up front could lead to issues later on if it’s not presented to the team before working on a item that needs it. In conclusion I feel that the Scrum-agile approach was the best method to use for this project.

References

Cobb,C,G. (2015). *The Project Manager’s Guide to Mastering Agile : Principles and Practices*

*for an Adaptive Approach*. Wiley. ISBN: 9781118991046