Sprint Review

Zach Behrensmeyer

Software Development Lifecycle

2/15/2023

During the review and retrospective there were points made that helped prove what each role did to help guarantee the success of the SNHU Travel Project. First, The Product owner made key communications with the business to get the vision for our project, communicated it back to us so we could plan using user stories, and communicated changes in requirements so we could adapt and meet the businesses new requirements. The team said it was my assistance through guidance in scrum events, and helping create the Agile team charter to follow helped them keep moral high and expectations set within the team. The testers were able to ask questions from the product owner through email and scrum event communication to write their tests and adapt to the changes that were made mid-sprint by altering their stories. The strength and coverage of the tests were vast and effective in helping us deliver this functionality on time at the end of the sprint. The developer was able to quickly react and investigate for a solution to the change in requirements the business communicated. Doing this we were able to keep the business happy with what was delivered after the small change in the requirements and were still able to meet the new sprint goal.

A good example of why the scrum-agile approach was good in getting user stories completed was that the product owner was able to listen to the customer and write user stories with acceptance criteria. These acceptance criteria are based on the direct feedback from the customer. The team was able to size these and estimate how much work they were able to bring into the sprint. This allowed the tester to write test cases that were based around the expectations of the customer. Based on the same information the developer can create software based on the ideas of the customer and the tester verifies that it does what it says it should. They are not needing to worry about what this will be a year out as we don’t care yet, so this allows the team to have a good work life balance as they can use the scrum ceremonies to size stories accordingly and can pull in work based on their capacity, the number of story points that can be completed in a sprint. Doing this allows the business to better understand what they should be expected to have delivered in a sprint interval. If they make changes, they should understand that if the team cannot handle the changes, they just must wait a little longer, or pull prioritized work out and replace it with the new requests.

If the project were using the waterfall workflow when the team was asked to change the focus of the website to be about detox/wellness vacations, it could have gone less smoothly than the team having a single meeting and saying that they will see what they can do to the existing functionality like it did in the SNHU Travel Project. By the time we were doing this work it could have completely thrown off the delivery date, especially if it was a lot more work. It would require a lot more meetings and lots of changes to existing documents. This is because with waterfall most if not all the requirements are defined before development happens. It is easy to forget thoughts about features while brainstorming initially that is one of the few reasons Agile is better is because it makes adapting to changes like this much easier. The team can take feedback from the business in scrum reviews. If something is wrong, they can tell the team and we can adapt and correct the software.

One good example of effective communication was when the tester sent an email to the product owner to ask questions about expanding the acceptance criteria and getting other information. The reason this is effective was because the tester asked the product owner for additional information to make the tests better. This would get everyone in the team involved if it is presented in a refinement meeting. This allows everyone to give feedback and get closer to the story so as a collective group everyone will understand what the functionality of the software should be. Another good example is when the developer wrote an email back to the PO and tester. This was effective in letting the team know what they could expect to be delivered as the sprint goal will need to change. The PO could let the business know what to expect after the sprint and the tester will know how they need to change their tests.

A good example of an organization tool that was used in the SNHU Travel Project was user stories. User stories introduced a lot of different options to the team to deliver the product. User stories allow the team to break the story up into smaller sized tasks, split the work up amongst the team, size them which allows them to see how much work they can pull into a sprint, and prioritize actions. The team could have used an information radiator such as Jira to move these tasks as status changes occur, such as To Do, Doing, Reviewing, and Done. This allows the team to have a place where they can check that should show what is currently happening with certain work. With Jira tasks can be placed in a backlog and sorted into sprints. This is nice because we can review our entire backlog at any time. We can review this to see when something is expected to be pulled in or see what can be moved around when something needs to be traded out.

I think that the scrum agile approach was best suited for the SNHU Travel Project. I feel like a travel site is going to require lots of change related to customer feedback. Agile thrives in an environment like this because the business can just keep taking feedback from their customers and work with the PO to create user stories for the developers to create new features. If they decide the priority on an item has changed it can be moved into the current or next sprint depending on the timing. There are two major cons to agile in my mind, but they are minor. Planning in the beginning can be difficult, and there are lots of meetings that are necessary. Features keep coming and the team keeps working on prioritized work, so it gets easier to work as the team understands the size of the work. The first issue is mitigated after time. The second one is more of a personal issue, but it is an unfortunate necessity, I don’t like being in meetings all day and sometimes that is my reality. This is something people must get used to when working with Agile.