Daniel Giannatsis

12/2/2021

Software Development Lifecycle

Sprint Retrospective

The various roles on the Scrum-Agile team worked together to deliver viable, sufficiently featured product. First, as the Scrum Master, I contributed to the success of the project by first assembling the team and scheduling all of the scrum events. I facilitated the scrum meetings and advocated for team members when needed. I also helped with sprint planning, especially for a new team like this. In addition, I helped the product owner with backlog refinement when necessary. Teaching agile and scrum principles to the team empowered them to code and test in sprints to shoot for faster working software and helped them adapt to and implement changes in requirements from the product owner. I also asked how the shift to detox/wellness vacations would affect the number of sprints to help estimate the time needed. Our product owner, Christy contributed to the success of the project by holding a focus group to understand the types of improvements / features the customer wanted the most, which became extremely valuable feedback that drove the software we ended up creating. She translated the requirements discussed into user stories, which convey the desired functionality, business value for the user, and acceptance criteria. She prioritized the user stories based on importance, since some user stories from different users were related and therefore more valuable. Later on in the sprint, Christy also communicated the shift in focus to detox/wellness vacations to the team so they could adjust and re-prioritize the backlog to reflect the change. Our tester Brian contributed to the success of the project as he was the one responsible for creating test cases for each of the user stories to be developed. He defined the inputs / user interactions involved in each user story and the measurable, expected results that the software should perform. He asked for more details from the product owner to help make the test cases as specific as possible. He also revised test cases when there was a change in the project, like when Christy the product owner decided to go with a slideshow page view for destinations instead of a scrolling page view, or the change of focus to detox/wellness travel destinations. Our programmer Nicole contributed greatly to the success of the project as she did the actual coding required to implement the desired functionality, the one responsible for frequently delivering working software. As the requirements of the project changed, such as the shift from list view to slide view or the shift to wellness /detox vacation content, she redesigned the pages by diligently changing the code.

The Scrum-Agile approach to the SDLC helped streamline the process and involved the whole team in bringing each user story to completion. Starting with the product owner’s focus group, the most critical user requirements were captured and translated into user stories with value statements that make it clear to all what the user wants to do, and the business value the user gains by doing it. Next, the user stories were further defined: the product owner included a relative size for each, estimating how long each may take. Acceptance criteria was also defined for each to make it clear what a successful feature looks like to the programmer and tester on the team. They were then added to the backlog where they were subsequently taken on by the team in the sprint. Next, the tester clearly defined a test case for each active user story, including the user interactions involved and the desired outcomes. The user acceptance criteria and clarifying information from the product owner helped him do that. With those defined, the programmer then worked on the features and had them verified to make sure that they were sufficient with what was desired. This use of test-driven development ensured that the completed features satisfied the requirements.

The Scrum – Agile approach supported interruptions and changes in direction as a meeting was held with the product owner and the rest of the team to communicate the shift to detox/wellness destinations and the relevant new requirements so that the team could swiftly and gracefully pivot in the new direction. The product owner made the transition easier by re-ordering and updating the backlog for the team. The time-boxing nature of the sprints allowed the team to have a fixed amount of time to complete as many of the updated user stories without pushing back the deadline. Definition of done was used to sufficiently complete each story without overdoing it to efficiently complete the most work in the allotted time.

The following email of questions from the tester to the product owner is a great demonstration of effective and collaborative team communication, where the email really helped dial down the specific criteria needed to create accurate, complete test cases for the project: To: Christy Subject: User Story Clarifications: Dear Christy, I have taken a look at your user stories and am developing test cases for the different features to determine whether the product passes or fails. I need a bit more detail so that I can use more specific metrics to clearly define my test cases. Can you answer the following questions for me? <specific questions for different users’ stories>. The plea for clarification helped the project immensely as the tests were now as true to the user needs as possible. Another great example of effective communication was when Christy the product owner de -escalated the worry present after the announcement of the change in project direction. She addressed the uncertainty and how the team would adapt to the change, spurring each team member to ask questions about their cross-functional role in the change:

Developer: What? Are you saying that everything we’ve done up to this point has to be scrapped?

Product owner: No, no, we just want the types of travel and vacations to focus on detox/wellness travel.

….. Since this is Agile, we want to keep the same dates, I will deprioritize other stories in the product backlog so that we can focus on this product. What can we get done in the time we have left?

Dealing with change, especially when one has already started a fair amount of work can make the transition difficult. In the case of our programmer, the team needed to get her to accept change and get in the right mindset for the rest of the sprint. The product owner did a good job here communicating what the change would entail to give the programmer an accurate estimate of the remaining work for the sprint.

The product backlog is an extremely useful tool as it lets the product owner break down a project into independent, descriptive tasks that are given a priority ranking and then assigned to a sprint, and also rearranged when the direction of a project changes. Next, the sprint planning meeting is important as it is when the team decides how many of the backlog items, they think can be accomplished in one sprint. This number of course becomes easier to decide based on how long the team has been working together with an established velocity. The Scrum daily standup meetings were useful as team members expressed their progress with the work completed yesterday, what they would accomplish that day, and things that were blocking them. Other important information could be side barred for further discussion between the member and the Scrum master. These daily meetings were helpful as the team was made aware of blockers early to maximize the workflow. They also informed the team of their current velocity. The Kanban board was useful as it clearly indicated items for each member, that were in progress, were blocked, and items yet to be worked on. Some Agile principles that helped guide how the project were:

“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage” This principle was clearly practiced, as changing requirements from the customer were dealt with swiftly.

“The most efﬁcient and effective method of conveying information to and within a project team is face-to-face conversation”. This was also practiced as almost all communication was face-to-face.

A first pro was the cross collaboration such as having programmers and testers on the same team which allowed for faster programming and testing resulting in rapid working software. Another pro was that changes in project direction were easily managed by the scrum-agile principles and tools. The team adapted to changes by updating the user stories to work on iteratively without changing the timeframe. This wouldn’t have been possible using the waterfall methodology. A final pro was that increased transparency, a lack of product manager, and a scrum master that encouraged the team to work things out amongst themselves lead to a more self-sufficient team that took responsibility for the whole project equally which led to more empowered individuals. I think that because of our more empowered team members, we created a much higher quality product. A con was sometimes the process, being one of frequent change due to uncertainty can put a lot of stress on certain members such as the programmer but that is something to get used to as embracing frequent change is the nature of agile.

Based on the aforementioned pros above, I believe that the Scrum-Agile approach was the best choice for the SNHU Travel project. This is most obvious since the project did have a significant level of uncertainty and therefore a more plan-based approach would have had significant time, resource and cost waste that would have had them possibly release a sub-par product that wasn’t really what the client was after. Scrum-Agile helped in many ways to fully realize the client’s vision for a great travel booking site that is relevant, easy to use, and valuable to the users.

Works Cited

Cobb, C. G. (2015). *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach* (1st ed.) [E-book]. Wiley.