Chada Tech

SNHU Travel Project: Sprint Review and Retrospective

David Vega

Southern New Hampshire University

CS-250: Software Development Lifecycle

Professor Haruka Konishi

February 18, 2022

**Scrum-Agile Team Roles: Contributions**

Throughout this course I assumed the role of various members of a Scrum team. The team was transitioning from a waterfall approach to an agile approach to develop an application for the fictitious client SNHU Travel which sought to expand their client base with new tool offerings for their customers. The team consisted of a Product Owner, Scrum Master, and Development team (which consisted of developers and testers). This paper seeks to provide an analysis of the scrum-agile methods applied to this project. For this analysis, I will need to assume the role of Scrum Master in order to draw conclusions about how these methods did or did not contribute to the final deliverable.

**Product Owner**

Every member of the team is a critical player in an agile project. However, it all starts with the Product Owner. This role is important because it is a direct link between the client and the development team. As the product owner, my responsibilities go beyond traditional project management. In this role, I had to define the requirements of how the project would be implemented. Part of the requirements came from the client direct while other input was collected from a focus group that I conducted with various end-users. I was responsible for creating and prioritizing user stories to add to the product backlog. These user stories would subsequently shape the way the development team would approach the project from start to finish.

**Scrum Master**

When I assumed the role of Scrum Master, I was tasked with supporting the product owner with backlog creation and maintenance while ensuring total transparency at all levels within the scrum team. Once the product owner defined the user stories, I would facilitate a sprint planning session with the team to review each of the user stories that would get accepted into the first iteration of development, otherwise known as a sprint. During the sprint planning session we implemented the estimation technique called planning poker. This technique helps the team to define the level of effort that each user story requires. Using the Fibonacci number system, we were able to assign a number to a user story that represented the level of effort based on a similar previously executed project. Since agile is new to the team, we did not have previous projects to compare to so we assigned a level of effort to the first user story and then based subsequent estimations on the preceding user stories. Once the backlog items were defined for the first sprint, project development began. I committed to a daily stand up meeting which is a quick fifteen minute overview of the day’s activities. The benefit of these stand up meetings is to maintain transparency and to identify and mitigate any uncertainty that could impact development. Each member of the team should be able to share what they did yesterday, what they will be doing today and what barriers are present that would prevent further progress. Once the sprint was completed, we gathered to discuss what parts worked and didn’t work in preparation for the next sprint.

**Development Team**

As part of the development team, I was positioned in the role of developer and tester. Both of these roles served as critical components to the scrum-agile process because in these roles, is where the business value is produced. As a developer, I was given access to resources that would help support my newfound role within the development team. These resources included day workshops that would allow me to understand and build on my knowledge of agile. In addition, I was given creative freedom to structure my code as I see fit using industry best practices. As a tester, my responsibility was to be able to collaborate with all members of the team to create test cases in order to identify any bugs that may be introduced. This is an important role because as the old saying goes, “Test early, Test often” is a key principle in iterative development.

**Scrum-Agile Approach to SDLC: User Story Completion**

The scrum-agile approach to the SDLC (Software Development Lifecycle), really helps to isolate critical functionality within a project. Software planning can be very complex if not executed properly. Having the ability to break down complex tasks into smaller increments is a path to a successful deployment. With the SNHU Travel project, requirements were collected from end-users and we created a succinct method (known as user stories) of defining the functionality of these requirements. These user stories are meant to be short but descriptive enough to be understood by users and developers alike. The standard practice for user stories is to state the requirement and isolate the functionality and its purpose. A user story consists of the *who?, what?* and *why?* The “who” represents the intended user, the “what” represents what the user needs to accomplish in order to complete a task and the “why” represents the reason behind the functionality which adds value to the requirement.

**Scrum-Agile Approach: Project Changes**

Agile by definition means *“supple”* and *“responsive”* and by that very definition is not immune to changes. Agile projects are expected to have some level of uncertainty. For example, the change in direction for the SNHU Travel project to focus on detox/wellness travel allowed us to take what was already developed and add/alter the code to support the new requirement. There was no production loss for this change in direction.

**Communication**

As a developer, communication is vital in weeding out bugs before code implementation. In order for me to complete a task, I will need to clarify rather than make assumptions about what I think the customer wants. This was evident in the developers communication to the product owner and tester as seen in the email below:

*To: Christy (Product Owner); Brian (Tester)*

*Cc: Ron (Scrum Master)*

*From: Nicole (Developer)*

*Subject: NEW requirements clarification and testing guidelines…*

*Hi Christy & Brian,*

*As per our discussion, I am beginning to revise the code to implement the new requirements for the focus on detox/wellness travel destinations.*

*If the original code base is not changing, then this new addition should be fairly simple to implement. However, Christy, would you be able to confirm with the customer if they would like to have the default display set to view detox/wellness destinations or would they like the user to select it in their traveler profile? Also, Brian, would you be able to provide me with some test case scenarios so that I can implement them in my code?*

*Thanks,*

*Nicole*

I feel this email was concise yet to the point. The developer does a good job in stating the requirement first, then asked for clarification from the product owner. She then asks the tester to provide additional testing requirements so that she can implement them in her code. There is also a calmness to the tone of her writing which suggests an willingness to be responsive rather than display a vehement feeling of resentment for having to accommodate the change. Depending on the product owner’s response, the tester may also want to create new test cases based on the requested information and submit to the developer. This creates an environment where transparency and collaboration intersect.

**Organizational Tools**

# Works Cited

**There are no sources in the current document.**