Chris Trimmer

CS-250-X6087 SDLC

8/10/2023

SNHU Travel Project

Sprint Review and Retrospective

**Overview**

ChadaTech has recently been hired by SNHU Travel to develop a travel booking web application. ChadaTech has been performing software development using a waterfall development model for many years. ChadaTech would like to explore software development using the Scrum framework because they think it will yield higher quality products and build a stronger corporate culture. They have decided to develop the SNHU Travel web application as a pilot to test and explore the Scrum framework. The following is a Sprint Review and Retrospective of this exercise.

**Scrum team roles and contributions**

Scrum is a framework based on agile software development. It works based on four foundational values, and 12 guiding principles that enable teams to develop in an evolutionary way – adaptable and flexible to the dynamic nature of business and the software industry. Scrum emphasizes an empirical approach to development, rather than the strict, linear structure of traditional models like waterfall. Development is iterative and performed using small teams of people with cross-functional skills. The experience of the team, observations of the current state of the project, as well as the dynamic nature of business requirements and market conditions influence how, what, and when features of the application get developed.

The framework includes a set of team members that include: a Product Owner, a Scrum Master, one or more developers, and one or more testers. Of course, the client and stakeholders are involved in the process, as well.

Client

The client for this project is the SNHU Travel President. The client is the ultimate driver for the project, defining the business goals of the product, and the type of product the business is interested in. In this case, the product SNHU Travel wants is a travel booking web application that will help them expand their customer base and remain one of the top travel agencies in the world. Customers are involved as part of the overall feedback loop. There were consulted during development to get their ideas regarding what they would like to see in the web application.

Product Owner

The role of Product Owner serves several key purposes. First, the Product Owner is the main point of contact for information flowing between stakeholders and the development team. She makes sure that the developers have the information they need to continue working, makes sure that stakeholders remain aware of the status of the project, and ensures that customer requirements are communicated. This leads to another key contribution of the Product Owner: managing the product backlog.

Since the Product Owner is directly communicating with the client, she can more easily prioritize the product backlog. An example of this during the SNHU Travel Project is when the client changed the requirement of listing the top-5 overall vacation packages, to focusing on the top-5 detox/wellness packages. The Product Owner communicated this change to the team and reprioritized the backlog so that they could meet the deadline of the overall project.

Scrum Master

As a comparison to traditional teams where there is a leader or project manager, the Scrum Master is more like a “servant leader” to the organization as well as to the development team (Cobb, 2015). She is not a task manager providing direction to the team. Rather, she coaches and leads the team through project completion by guiding them through the agile process. She is not responsible for getting the actual work done, rather she is responsible for ensuring the team completes the work in an agile way. She removes blockers, carries out the Scrum events, moves the team toward self-organization, and overall helps the team focus on creating the work product during each sprint. The Scrum Master carried out this role effectively throughout the project by coaching the team, removing blockers, and defining and describing the Scrum process when it was necessary to do so.

Development Team

The developers are responsible for turning user stories into functional software. This includes determining the best way to implement a user story, performing the actual programming and configurations, as well as communicating with each other and other Scrum roles to ensure everybody is aware of their status. In the Scrum framework, the developers self-organize – collaborating to determine how best to complete user stories. This occurred throughout the project as code was completed and updated during each iteration. The code was well documented and followed industry standards.

Testers

Testers are extremely valuable to the Scrum process. First, testers are involved throughout the development process during the sprint, as well as the planning process for each sprint. Second, testers are continually testing during the current sprint, and continually improving software during each iteration. Third, testers are tightly coupled with developers and the Product Owner to ensure features are being implemented based on expectations of users and final product. Finally, testers are adaptable and flexible to change based on priorities and the ongoing changes in each iteration of development, and therefore perform regression testing, as well as user acceptance testing. During this project, the tester performed all these responsibilities.

**Scrum approach to completing user stories**

Scrum has three empirical pillars that guide and support the five key sprint events. The pillars are defined as transparency, inspection, and adaptation (Schwaber & Sutherland, 2020). Each pillar supports or guides how the development team carries out the sprint events. The sprint events include the sprint planning meeting, daily-Scrum, the actual sprint, the sprint review, and the sprint retrospective.

Transparency is the process of making the status of work visible to everybody involved in the project. Furthermore, there should not be hidden agendas, nor concealed information about the project or its status among team members and stakeholders. This is consistent with the core value of openness in Scrum. Openness and visibility of tasks associated with the project were accomplished in the sprint via the Scrum events and using the Jira tool.

Inspection is the process of evaluating, inspecting, and reviewing the Scrum artifacts and the overall progress of each sprint. Scrum artifacts include the product backlog, sprint backlog, and the increment. During each sprint review, stakeholders and the development team inspected the deliverables for quality, functionality, and asserting if the product is as expected.

Finally, adaptation is based on performing course correction as soon as possible if something is not going as intended, or if confronted with change in requirements. This might include determining that there is a defect, that the product is operationally unacceptable or is poor quality. Since Scrum is so focused on communication and collaboration, adaptation is something that the development team is empowered to do because they are self-organized. Adaptation was accomplished during the sprint retrospective, as the team can course correct in preparation for the next sprint.

**Adaptation to change**

A benefit of using the Scrum framework is the ability for a development team to respond to changing requirements. During development of the web application, the development team was informed of a major change in requirements. The original specification was to build the web application to support displaying the five top vacation destinations. The team wrote test cases, began development, participated in Scrum events, and were committed to the artifacts during each sprint. SNHU Travel changed their mind and preferred to focus on detox/wellness vacation packages.

Using the Scrum approach, the team was able to continue with development through project completion even though they were confronted with this change. This was accomplished using the Scrum framework in several ways. First, work is accomplished in sprints, which are defined time periods. The project starts out with a main product backlog of prioritized user stories. Within each sprint, there is a sprint backlog that contains a prioritized list of user stories to be completed for the current increment. User stories are prioritized and completed based on their value to the project at that time. During each sprint, the features and product are inspected and validated by team members and stakeholders.

Second, the work that is accomplished during each sprint are defined using user stories, and they are prioritized in the product backlog. User stories are requirements told from the perspective of a user (or role), and do not contain detailed technical information. This enables stakeholders and team members to discuss the requirements, see their status, and determine priorities more easily.

As previously defined, the third pillar of Scrum is adaptation, which is directly related to what happened during development. The team was confronted with a major change in direction, and they were able to adapt. The Product Owner quickly gathered the team and informed them of the change. The development team was self-organized and therefore empowered to figure out how to implement the change without outside interference or being told how to do it. Furthermore, team members were free to communicate with the Product Owner via email or follow-on discussion to gain clarifying information. Finally, the team was able to reflect, and adjust processes to help make the next iteration successful.

**Communication**

Communication occurs naturally in the five Scrum events - daily Scrum, sprint planning, sprint review, sprint retrospective, and meetings and discussions that occur throughout the sprint such as during backlog refinement. Communication is an integral piece of transparency and openness in Scrum. It works because all team members are involved in the discussions, and the team can self-organize based on the discussions. Furthermore, communication and Scrum events promote the core value of openness.

The team also communicated via an information radiator using Jira. (Jira will described in the next section.) The value of an information radiator is that stakeholders, and all team members can always maintain situational awareness of the project and its associated user stories, without having to physically meet. It is like a continuous update what is happening in each Sprint. As described earlier, face-to-face communication was accomplished in the scrum events which was effective at discussing how work was being accomplished, and it enabled developers to help each other solve problems or resolve blockers amongst themselves.

**Scrum tools**

As previously discussed, the development team utilized Jira to manage the project. During the initial sprint planning event, the Product Owner created a ‘Project’ to define and represent the whole SNHU Travel development project. Within the Project, she then set up ‘Epics’, which represent the high-level requirements of the project and are a collection of user stories. Each user story is associated with an Epic. The Scrum Master also created a kanban board, which served as a visual, information radiator where all the user stories could be seen in their respective states.

The development team had the ability to transition a user story from one state to another based on the status of their work. For example, the visual representation of a user story transitioning from ‘in development’ to ‘in test’ is a simple matter of dragging-and-dropping the user story from one column (or swim lane) to another. This visual representation being centrally located helped stakeholders and team members always remain aware of the status of the project. Furthermore, it promoted discussions about the project, and enabled everybody to make more informed decisions.

The ability to visualize the current state of the entire project via transitions and workflow states is what makes Jira such a powerful agile management tool. Teams can quickly and easily communicate change in a visual way; the tool promotes communication because everybody on the team looks at the project, epics, backlog, user stories, and kanban board in a centralized location – they know the status at any given time. The result is that they can adapt and react more effectively to dynamic and rapid change, as well as make better decisions.

**Effectiveness of Scrum**

The Scrum approach was the best approach for the SNHU Travel development project. A primary basis for this assertion is that the team was confronted with two changes in requirements in just this short five-week timeframe, and they were able to adapt and showcase a finished product. Furthermore, the product was of high quality and met the requirements and expectations of stakeholders. There were no negative impacts or cons associated with using the Scrum framework for this effort.

A primary reason the project was successful is because using the Scrum framework enabled the team to work empirically. They were able to self-organize and make decisions based on their experiences, combined skillsets, and openness of communication among stakeholders and other team members. Scrum works by developing tangible software in fixed timeframes (sprints) using small teams of people who have the combined skillsets to accomplish the project. This enabled the team to plan the work that could done during each iteration and adjust as necessary to maintain the pace of development.

Another benefit of using Scrum is that testing was performed throughout the sprint. It cannot be overstated enough how valuable testing is to the Scrum framework. In a Scrum approach, the tester is involved with the development team and stakeholders from beginning to end. This means that the tester’s perspective is being communicated during sprint planning as well as during the sprint and the final stages of the sprint. This essentially enables acceptance testing in each sprint, as the feedback loop among stakeholders and customers is applicable as the product is being developed.

Finally, we should discuss documentation compared to most traditional development models. This includes modelling such as use-case diagrams, UML class diagrams, and flowcharts, as well as requirements specification documentation that detailed functional and non-functional requirements. Scrum intentionally reduces the amount of documentation of the system, and instead developers are empowered to collaborate with each other and develop based on how to accomplish the user story. In lieu of documentation, the team used Jira to ensure stakeholders and team members had constant situational awareness of the project.

SNHU Travel needed this web application built in five weeks. It would have taken valuable development time to produce the amount of documentation required to fully define and model the project. Furthermore, consider the change that was made during development. The team would have been required to re-do or update the existing documentation in addition to performing the changes. The maintenance and upkeep of documentation can sometimes be more cumbersome than the value it is worth in implementing the project.

The SNHU Travel web application was a pilot for using the Scrum framework and it ended up being a success story. The team was able to develop an application that produced value to SNHU Travel in a short period of time despite two major requirement changes during development.

**References**

Cobb, C. G. (2015). The project manager’s guide to mastering agile: principles and practices for

an adaptive approach. John Wiley.

Schwaber, K., & Sutherland, J. (2020, November). Scrum Guide | Scrum Guides.

Scrumguides.org. https://scrumguides.org/scrum-guide.html