Final Project: Retrospective

Blake Condrey

CS 250, Software Development Lifecycle, SNHU

April 16, 2022

Final Project: Retrospective

Throughout CS-250, the course offered the opportunity for students to undertake the various roles of a Scrum – Agile team that would exist in enterprise in the real world; namely, the Product Owner, Scrum Master, Developer, and Tester. Living in these roles through each of their respective weeks in the term provided the scope and responsibilities *of*  the role, and simultaneously, its interactivity and dependency of the other roles, providing a realization on how the team operates and more notably, how an scrum – agile team is remarkably successful in the real world with shipping world-class software products to clients. In my experience, accepting these roles and responsibilities, in relation to the SNHU Travel Project, contributed to the success of the SNHU Travel Project, specifically by outlining the communication between the roles.

For example, the Product Owner and Scrum Master communicated directly to the client, extracted useful feedback to build user stories, which, when wearing the developer’s hat, outlined the definitions of acceptance criteria and the “what” of the product to be designed. Later, as a tester, we were able to step back in the stories and highlight limitations to the stories and derive cases that could expose bugs or development shortfalls to be remedied, all to ship a successful end-product. The user stories proved to be a powerful tool throughout this process, as it centralized points of communication to tangible items, and defined the product in consumable components that simplified the product scope. This approach is rather typical in the Scrum – Agile Software Development Lifecycle, and highlights the flexibility and adaptability, as the project took an abrupt turn to shift to a new idea.

Initially, the core task, based on the client’s user feedback, was to compose and render a curated list of travel destinations that met the end-user’s desires as defined in their profile. For instance, if one user prefers beachside locations, their curated list would most likely omit travel destinations that were in the mountainous ranges or national forests, and vice-versa. As the project grew, new feedback came from the communication streams, that SNHU Travel would pivot to marketing detoxifying and wellness retreats. Through user stories provided – that were pointed to scale of difficulty – developers, testers, and designers had the ability to shift directions quickly and maintain the goal of the release date. In this experience, I was able to determine the importance of the iterations approach, as in the case of a waterfall approach outside of agile, this may not have been possible at all, as likely, an end-goal would have been previously defined, which in no way would have related to this feature, thus having no scope to requirements, potential side-effects, or the ability to communicate with users during the shift.

In Week Six, we were grouped into a simulated Scrum – Agile Team, where each of us were tasked with choosing one of the various roles and undertaking the responsibilities related to the respective role. In this exercise, I decided to become the team’s Product Owner, and made quick decisions to outline the team’s path, while also choosing specific project management tools to ensure that we would have the resources needed to effectively ship a meaningful product free of encumbrances. For example, I noted that, in tandem with the Scrum Master, we would immediately implement daily stand-up meetings, where each member would outline their current position in the following manner:

* What did you accomplish yesterday?
* What is your plan for today?
* What blockers or impediments are halting your progress?

Also, the team would begin to have sprint meetings where, using Kanban boards, user stories, dev chores, or other various tickets would be outlined and presented to the team. The developers and testers would then simultaneously “score” the ticket – a tactic known as “story points” – to give the team an understanding of the ticket’s complexity, scope, or difficulty – a metric to be determined by the team – and move that ticket into a new column on the Kanban board from “IPM” (iterations planning meeting) to the “To Do” column, or potentially if the score is rather higher than most, discuss how to separate this ticket into smaller components. The Kanban boards would exist in GitLab, along with our code repositories, to handle multiple members actions simultaneously, as well as pushing our “master” code branch through continuous integration with the staging environment exclusively available to all members of the team, and most notably, the users, to gain immediate feedback to build new user stories or reiterate on previous stories and adapt as needed. The Lead Developer on the team was instrumental in highlighting the need to have a quick meeting with her and myself, along with the Scrum Master, to predetermine any changes to the tickets as needed prior to introducing the product backlog to the remainder of the team; she would speak on behalf of the developers first, saving potentially hours-to-days on discussing the ramifications to the product that each ticket may impose. Unequivocally, these communication pipelines that we created, using the tools and resources that I chose, would have certainly proven effective in a real-world scenario where a multi-disciplinary, flexible agile team would exist and can build a meaningful and extensible software tool for their clients.

Throughout the process of implementing the Scrum – Agile methodology on the SNHU Travel Project, several pros and fewer cons can be extracted and discussed upon. Notably, the chief advantage was immediate flexibility and adaptability. Given that agile iterates on the SDLC throughout sprints, we were able to easily communicate with the users, which would have sped up the time to market release. Also, this approach proved most effective in the access to communication across the board. The Product Owner/Project Manager worked intimately close with the developers, and while they do not directly produce code, they steer the direction of code based on continuous and targeted communication with the client – a process not inherently available in the waterfall methodology. This direct communication created ownership and pride amongst the team, as each member was able to see their immediate impact throughout the iterations, improving overall morale across the board. If a con can be extracted, I believe it existed in the immediate pivot; it likely affected morale in knowing that the entire codebase would have to be altered and left the developers wondering where the issues existed. Most likely, however, the team can reestablish position knowing that overall, the end users will be happy with the final product.

Throughout the course, I can undoubtedly posit that the agile framework would have been the only effective approach in shipping SNHU Travel for several reasons; namely, a short window of time of release to market, and the immediate pivot that was encountered. In a waterfall approach, this simply could not have been accomplished, as there existed more than one distinct goal to be accomplished, and flexibility and adaptability were proven to be paramount.