# CS-250 7-1 Final Project

James Furman

james.furman@snhu.edu

Southern New Hampshire University

CS-250-T3288 Software Development Lifecycle

Prof. Haruka Konishi

February 19, 2023

Christy, the Product Owner, fundamentally guided the project by creating the product backlog. She held an interview with a focus group of SNHU Travel’s top users to determine how value could be created for the company and their customers, and then used that information to guide her formation of the backlog. Additionally, she stayed in ongoing contact with SNHU Travel, facilitating an update in requirements when they determined that they wanted to focus on detox and wellness-based retreats. She communicated the change in requirements in both an impromptu meeting she called and in modifications to the backlog items. In doing all this, she created and maintained the backbone of the first Sprint in the forward-looking and adaptable manner Scrum calls for, which will facilitate the creation of more stakeholder value in the same SNHU Travel project going forward into our second Sprint and on.

Ron, the Scrum Master, played a part in Christy’s impromptu meeting by requesting clarification about whether the new requirement implied changes to their deadlines or timeline. But compared to the other three Scrum team members, he was conspicuously low-profile in meetings and artifacts – and intentionally so. As a Scrum Master, his focus is on leading by example, getting everybody on board with Scrum, and removing impediments to the development team’s work of creating stakeholder value. This does not necessitate participation in meetings and in fact, since the team performed well in this Sprint, Ron’s relative silence points to good work on his behalf and especially points the team doing a good job learning Scrum. (It would be a very different story if the team had performed poorly – then the Scrum Master would need to step in and abdicates his duties if he fails to.)

Nicky, the Developer, rapidly prototyped the first user story, the slideshow. She then responded to the change in requirements by requesting information about the change’s implications for her current work, and by updating the prototype. And Brian, the tester, developed test cases for the first user stories, requested clarification from Christy about several aspects of those user stories, and revised the test cases based on her responses.

The central practical way in which the team’s Scrum-agile approach facilitated completion of the user stories was by focusing the team on and facilitating communication lending to the formulation of valuable user stories – not just features by another name. These user stories communicated requirements to the developer and tester, which allowed them to effectively estimate the size, and communicated the stories’ priorities to them, letting them rapidly iterate a prototype for the first user story and define “done” in technical terms for the first several stories. For example, the requirement to implement a search feature that users can use to find vacation destinations was implemented in bare-bones fashion, with forward-compatibility for search filters that can be created modularly; the decomposition of the general idea of a vacation package search into one user story for the basic search, and any number of future user stories for search filters, allowed a useful search function to be implemented far more quickly than if the search function plus a number of filters were all viewed as a single story.

Because completing a user story doesn’t mean checking technical boxes, but rather means creating value for the stakeholder, the Scrum-agile approach also helped the first user story come to *true* completion by facilitating adaptation to the project’s interruption and change of direction. The fact that the project had been formulated in user stories of limited scope allowed the change’s effects to be quarantined; the clarity and specificity of information in the user stories made it possible to recognize which user stories were affected by the change in direction, and how. It put in place ongoing measures of communication but, perhaps more importantly, the commitment to using a Scrum-agile approach psychologically established the expectation of change, so that in a situation where change became necessary, everybody was able to effectively respond to that necessity. So, while some team members were initially perturbed and thought their work so far might have been rendered null, the team was able to get on track quickly, redo the small amount of work that needed to be redone, and appropriately adjust their expectations for the work they had remaining.

After creating preliminary test cases, Brian emailed Christy requesting information about the user stories to make sure his test cases would ensure the completion of the user stories. He communicated effectively by sorting his questions by user story and dividing them with bullet points, which made it graphically clear for Christy how many questions he needed answers to and what they pertained to. He also asked very specific questions, and put them into concise terms, including relevant explanations or examples to make sure Christy understood precisely what information he was requesting. For example, he asked, “When the user has made a search, what should the search results look like? Ex. paneled square thumbnails, horizontal displays, list of titles that dropdown when clicked – what appearance do you have in mind?” Here he put his question forward briefly and gave examples, to both 1. make it clear that he meant “look like” visually, and 2. give Christy some likely examples to make it easy for her to answer him.

After the change in requirements for the project, Nicky sent an email to Christy and Brian both giving and requesting information. She led into her email by briefly stating its purpose: “I want to communicate re. the new requirement Christy brought us, so you two know what I’m doing going forward and so I can make sure you’re up to date and I’m meeting your needs.” She also divided her email into sections, one directed to Christy and one to Brian, but put both into one email so that they could all be kept in the loop together.

Our team succeeded by using email, Word, and our software for creating and sharing user stories. As shown, email proved valuable in facilitating communication. We did not use it to prepare for any of our Scrum events, but we did use email to send materials and info that we had discussed during the Planning meeting and during some Daily Scrums, as well as for the various communications that are necessary in between events. We used our user story software for Christy to easily create effective user stories, clarify them based on discussions during the Daily Scrums, and for the other team members to reference them during their independent work. And we used Word to record the minutes of our Planning and the Daily Scrums, as well as Christy’s focus group with SNHU Travel users and her impromptu meeting with the rest of us when our stakeholder’s value definition changed.

There were a couple of cons to using the Scrum-agile approach. First, the implementation’s scale is small; only one team is using this approach, and problems are far less likely to crop up or become visible at such a small scale than when the whole organization is using it. Therefore, going forward, we need to remain alert about the Scrum-agile implementation so that we aren’t blind to problems it causes as they inevitably occur, rather than allow rose-tinted glasses to inure us against circumspection. For instance, this team did remarkably well understanding the Scrum-agile, but we should not be surprised if some people need more help to understand and apply the principles.

The approach was also significantly beneficial. The team was able to effectively complete the beginnings of the SNHU Travel project – but I do not believe that that was a benefit of the Scrum-agile approach. Frankly, I think it’s ambiguous whether the SNHY travel project was best approached with a Scrum-agile framework. The project is sufficiently limited in scope and run-of-the-mill that I am convinced it could have been completed in a waterfall-style approach with just about equal efficiency. Even the requirements change in the middle of the Sprint could have been adapted to by all but the most stiff-necked waterfall traditionalist. And the second con to our use of the approach is that, because of the project’s limited scope, some of the communications and artifacts we spent time on seemed redundant – “Couldn’t we just as easily make this site without sending all these emails? Without writing out all these user stories? Without quizzing the Product Owner about all these test cases?”

But there is one caveat to that assessment, which is that a waterfall approach does not center around creating and sharing knowledge in the team. Therefore, while a waterfall approach could have been applied equally efficiently here, a team using a waterfall approach would also be likelier to make mistakes resulting in lesser efficiency. And therein lies the real value of the communications that sometimes seemed redundant, and the substance of the Scrum-agile approach’s benefit: shared understanding. Everybody understood what was going on, at all times. When there were questions, they became visible rapidly and were answered promptly. Everybody understood what work he or she was doing, and why. Everybody knew what questions to ask, which isn’t something that will necessarily happen in a waterfall framework where team members aren’t empowered to self-govern. As I mentioned above, the simple fact that everybody had committed to using a Scrum-agile approach meant that the team was psychologically prepared to adapt. A small hiccup like this Sprint saw cannot sufficiently demonstrate the power in such a psychological shift, and what it did to the team may only be expressible by experiencing the atmosphere of the workplace, and not in writing.

I am convinced that the shared understanding of the project engendered by the Scrum-agile approach we implemented here caused the team to be more capable and adaptable than would be possible in a waterfall approach. And if we scale that, we will see a transformation of our organization’s ability to accomplish projects. I look forward to making it happen.