Sprint Review and Retrospective

Is an Agile approach better than a traditional approach for developing software? To answer this question, my team used a well-known and popular Agile method called Scrum to develop the SNHU Travel project. The team experienced significant advantages with Scrum in their ability to respond to changing requirements without losing momentum on the project, and in their ability to communicate with and support each other via Agile tools. What follows will be a summary of this approach and an evaluation of its effectiveness in contrast to a traditional, or “Waterfall”, methodology.

What distinguishes Scrum from other Agile methodologies are the emphasis on the sprint and the structure of the team.

The “sprint” is the term for the time-boxed period (usually two to four weeks) during which development occurs. Before the sprint begins, the team decides together what features of the project they will complete during the sprint, and after it, they talk about what lessons were learned, what changed, how they succeeded and how they can improve for the next sprint. This continuous cycle of setting short-term goals, working toward those goals, and reflecting and adjusting allows the team to continue learning and improving as they work, without sacrificing momentum or significant amounts of progress if things have to be changed, which they always do. In a traditional methodology, expectations and requirements are established at the beginning, often to minute degree, with no expectation of change built in. The sprint not only allows for change, but expects it, and sets an expectation for the team to adapt and improve, instead of to lock on to their assignments like a train car on a track.

The team is also agile. There are very few assigned roles, which means that each member should be capable of handling any task that needs to be done in order of value, rather than wasting time waiting to be able to do their “job” and then passing the proverbial ball to the next person, assuming their work is now finished. In core Scrum, there are only two specialized roles – Scrum Master and Product Tester – and all other team members are Developers. In order to make the transition slightly less of a shock to our developers, we added the role of Tester. In practice, however, our Testers and Developers worked closely together, which often doesn’t happen in a traditional development process.

The role of Scrum Master turned out to be similar to that of a Project Manager, but it required far less hands-on monitoring of progress and resolving efficiency and resource problems, because the team was empowered to solve most of their own problems on their own. At first they were tentative about taking ownership of the project and their own roles on the team, but once it sunk in that it was up to them, they out-performed themselves! From then on, the Scrum Master was able to focus on facilitating the needs of the team, coaching them in agile practices, and acting as the point-person with upper-level management.

The role of Product Owner has a few things in common with Project Manager, but is closer to a Business Manager. They are the point of contact with the stakeholders of the product – in this case, SNHU Travel and their users – and are responsible for making sure that the project delivers the best value. To this end, they take responsibility for organizing one of the most important tools in Scrum, the Product Backlog. This is an ordered list, visible to the whole team in a shared workspace, of the requirements of the project, arranged from highest value to least. Highest-value requirements take precedence and are developed first, ensuring that the team delivers the most value possible to the customer within the time dedicated to development. Our Product Owner did a fantastic job with responding the stakeholder’s needs and concerns, and communicating changes and feedback to the team.

The contributions of Developers and Testers should be fairly obvious: they developed the product. Rather than each specializing in a particular area of development, each member of the team worked on each task in order of value. In that way, we cut down on wasted time and labor. Developers and Testers worked closely together, and because of this, each became more comfortable with the skills of the other.

The flexibility of the team’s roles is extremely useful in adapting to changing requirements for the project, as we saw first-hand when SNHU Travel decided that they wanted the software to have a focus which was not specified before development began. It was freeing for them to know that they could make this request of the team, since the expectation of change had been laid out beforehand by the Product Owner, in accord with the Agile principle of communication over contracts. The Product Owner communicated this change to the team, and the developers were able to quickly adjust the necessary code to reflect the changes without losing much time. In a traditional project, even asking for a change in requirements can be a source of tension, not to mention to huge strain on the development timeline as the project is shipped around to various developers – who, in a traditional software development company, may already be working on a completely different project where their specialization is required – and it can set the development back significantly. The freedom to respond and implement change quickly was a boon to the team and convinced many team members that and agile approach is better than a traditional approach.

It may be clear by now that one of the key features of Scrum that makes it more efficient is good communication practices. As a part of Scrum, the team was collocated at work. Because of this, they were able to share ideas, ask for and give help when necessary, and quality-check each other’s work as they went along. Even when they were not physically in the same place, they maintained consistent communication about progress on tasks and to resolve interpersonal conflicts (and there were some; no new team forms without going through the “storming” phase). Below is an example in the form of an e-mail from a tester to the Product Owner, asking for clarification on requirements for a specific user story:

Hello (Product Owner),

I’m working on the test case for user story number #3, “As an end user, I want to be able to click on a link to view the top 5 most popular destination that match my profile or travel history, so that I can find vacations that match my interests.” Since we’re focusing on the Top 5 Destinations link function right now, I was wondering if you could give me some guidance as to how you envision the “preferences” applying to the search. Accessing the profile for user history or a profile setting seems like its own user story, or possibly an epic. Maybe for now we should make it a search parameter, to stay focused on how it affects the list?

I’d love to hear your thoughts about this. Thanks for your time!

-Tali

Even when communication was lax, the entire team participated every day in the Daily Scrum, which is a fifteen-minute stand-up meeting in which each team member shares what they’re achieved since the previous day, what they intend to accomplish that day, and what obstacles they expect to face. This meeting not only keeps everyone abreast of overall progress, but gives everyone an opportunity to reach out and ask for or offer support.

Another key tool to the successful implementation of Scrum, besides these daily communications, was the information radiator/Kanban board. As this was an experiment for ChadaTech, the team did not use a new online tool or software. We kept track of the Product Backlog and each team member’s progress on each specific task on a large board in our workspace, and this is where we met every morning for the Daily Scrum. It was also the center of attention for the Sprint Planning Meeting and Sprint Retrospective. This simple tool was extremely effective for the purpose of the keeping everyone up-to-date, as everyone saw whenever anyone got up to move a task along the board. The team found this not only useful, but highly motivating. A sense of teamwork and camaraderie was built up as the whole team was able to celebrate each small step together. We recommend, if Scrum is to be implemented company-wide, investing a high-quality software tool that replicates this type of board, since every team can’t always be physically in the same space every day, and many of these software programs come with built-in tools for breaking down team velocity and other useful features. We also recommend the use of the simple white board for the real-world sense of accomplishment it gives.

Overall, the SNHU Travel development team agrees that Scrum is preferable to the traditional development methodology. It has significant advantages in terms of flexibility and efficiency, and the sense of ownership and accomplishment that each member felt at the conclusion of the project cannot be overstated. While some members were uncomfortable with the lack of clearly defined requirements at the beginning of the process, we all agree that the trust and collaboration that developed between team members and between team and customer made up for that, and in the end each member was glad to have the freedom to adapt, communicate, and make mistakes so that they could learn from them.

A final word: this particular project was fairly small, and many requirements were actually fairy well-defined at the beginning. A traditional development approach probably would have been efficient enough without risking too much waste of time or resources, even factoring in the change of focus mid-way through development. However, we have concluded that the agile approach was more efficient in spite of this, and would certainly be preferable for other projects in the future, especially large ones.