Ryan Mitchell

CS 250

April 18, 2021

Sprint Review and Retrospective

As the interim Scrum Master, I must first take this opportunity to commend my team for all the effort they have put in, not only towards the completion of our application, but towards adjusting to this new system. We were tasked as guinea pigs to metaphorically dip our toes in the water and test out how the agile methodology might better improve company efficiency. The relative success of this project could not have been what it has been without each individual member of our team. The SNHU Travel project was our first benchmark to really see how prepared we were to make this transition, and the results looked very promising. The Product Owner did an exceptional job of communicating with the customer over the desired outcome of the product. She not only successfully curated a list of functions and tasks to be implemented in our design, but collaborated and received input from myself and the team as to how best to prioritize each story. I, the Scrum Master, did everything I could to organize my developers and testers and provide my own support both emotionally and directionally. I monitored progress, stayed behind after Scrum meetings to answer questions, and worked with the Product Owner to help create our Product Backlog. While I may have done everything I can to support my team, the developers and testers really do deserve so much credit for how well they were able to collaborate amongst themselves, and with me and the Product Owner. They were enthusiastic, dedicated, honest, and gave constant updates on their progress and expectations. Together, we formed a cohesive unit that I can only hope will be replicated as the company makes the full transition.

The difference I have noticed in overall team morale and satisfaction is tremendous. By dividing every feature into parts, and creating our User Story Backlog, it turned what could have been an overwhelming task into many very reasonable tasks. Unlike our usual waterfall method of planning before developing, this method not only keeps everyone engaged but also saw increased production time and less problems to be resolved. At every meeting, we discussed which features were best approached by whom, and saw collaboration between developers of certain strengths and developers who were struggling on certain topics. If user stories were unreasonable or too much for any one person, we were able to work together to negotiate how best to rework it and saw incredible growth in our team. Having daily/weekly meetings also promoted team chemistry, and the time spent away from the desk was clearly improving morale.

When we were faced with a major course correction, frustrations were indeed apparent. However, I told the team that this was all a learning process, and also part of the system. While the Product Owner will improve over time and elicit more refined features from the customer, this sort of issue would have been even more exacerbated in a waterfall model. I said that the fact we didn’t have to scrap everything and start over was already an improvement, and I think many understood. Luckily, when the break happened and we were forced to adapt our plan, it is thanks to the agile idea of constant check ins that we were able to address this shift immediately. Since only specific features that were incorporated in our teams user stories needed to be reworked or scrapped entirely, it helped us move towards a final deliverable with much less resistance.

Everybody who I have talked to about this system, I have mentioned at least once how communication was paramount. When our team was first created, I made a point of stating directly how important it was for them all to contact me with any questions or concerns. I realize that certain members were less open than others about issues, so in the initial stages of Scrum meetings I made sure to have every single person share their current progress, their projected progress, and one obstacle they had faced. That way, I could gauge based on the reluctance with which they answered the final question how much any person would be struggling. When problems were shared, other developers were even given the chance to provide their own potential solutions. When this still didn’t resolve the issue, I made a point of talking with the person face to face. It was through these methods that I was able to not only communicate with the team, but also allow the team to communicate with themselves and open doors to ideas that may have otherwise stayed shut.

As I mentioned before, communication was the major tool that Scrum and the agile system depend on. Scrum meetings, sprint planning, and sprint reviews were all tools that separated agile from waterfall. The constant feedback feeds into the idea that all of these independent moving parts can successfully operate in unison. By meeting frequently, and planning our sprints accordingly, we were able to meet deadlines to our near maximum potential. Supplementing these communicative practices, we also implemented the use of software such as Jira and Azura. This allowed the team to communicate visually on the progress of user stories in development, and also provided a direct messaging service to facilitate quick responses. This cut down on the number of emails which starts to pile up over time and improved communication even further. When our sprint planning was completed, we were even able to directly copy this into the storyboard format on Jira, so everyone had access to all we had discussed. These two practices combined into a very effective machine of consistent software development.

So what is my final verdict on the Scrum-agile approach? I think this system was truly built for where our company is heading. By beginning development on the SNHU Travel development project immediately and not strictly dividing time into blocks, we were able to successfully reduce overall stress and overtime as we approached our final deadline. By having increased communication, it also helped facilitate new bonds and better chemistry amongst our team. When issues arise, the agile approach allows us to shift instantly and decisively with little impact on our routine. I will say, however, that this system does offer a lot to be taken advantage of. By appearing more relaxed, some developers may not work with the same level of rigor, and tasks may not always be divided up fairly. Inevitably, someone will feel they got the short end of the stick and conflict can arise. It is my job to be reassuring, and look for complacency from developers and testers. That being said, given everything that has happened, I believe the Scrum-agile approach was the best approach we could have implemented. The world of software development is no longer a one man job, and with software becoming increasingly complex and layered, the need for teams is greater than ever. I think, given time, all parties will prefer it and I think the results will speak for themselves.