Kate Wheeler

CS250: Software Development Lifecycle

Module 7: Final Project

The idea of being Scrum Master for our company’s first attempt at utilizing Agile methodology has been incredibly daunting; the team, including myself, are all new to programming and prioritizing in this way. Despite this inexperience, our first sprint has gone exceptionally well. We have achieved client goals faster and more completely than ever. Our product owner was especially helpful when communicating with the customer regarding specifics of the projects, as the developers’ idea of what a “list” would look like ended up varying significantly from what the customer had in mind. Meanwhile our developers were able to meet these well-articulated goals within the required time frame with ease, and creating a functional initial version of the list’s code from the beginning meant that the changes to be made were minor to make a functional list that also pleased the customer (the slideshow). The testers did thorough work, ensuring that every aspect of the program is fully functional and accurate. Collaborating as a team to make basic decisions about our first sprint was uncomfortable, as we are all clearly new to working this way and each have different communication styles. Despite this slight awkwardness, we made some decisions. I’m confident this kind of decision-making communication will improve as we all get to know one another and gain the experience necessary to have confidence in our opinions on these matters. As Scrum Master, I planned our sprint based on these decisions, making sure to build in time for our spring planning meeting, daily Scrum meetings, and a sprint closing and retrospective. I also facilitated these meetings socially, encouraging everyone and making sure each individual had a chance to speak uninterrupted during each meeting.

Use of user-stories to plan the specifics of tasks during this sprint was extremely helpful for both the product owner and the developers. The product owner was able to be clear and specific about the expectations of customers, setting the developers up for success, as they were able to create a product that was precisely what the customer expected with only a couple tries. The improved stories provided by the product owner also make handing interruptions caused by their altered expectations much easier, as the developers didn’t have to take the time to replan as they would under waterfall coding circumstances: when alterations were made to the user story regarding the style of list, the developers simply read the new user story and started implementing the specific goals from that story instead.

Communication was an area in which the team particularly excelled during this first sprint. Though our pre-planning forum-style chat was a bit awkward, utilizing email, messaging, and in-person meeting time rapidly increased team rapport. During in-person meetings, I maintained each person’s ability to speak uninterrupted and encouraged engagement from the entire team. One method that we used to give every person a chance to contribute was planning poker, in which we look at tasks and each choose one of several values to represent the time and effort that they think would go into each task. If everyone didn’t agree, we gave each person who felt passionately a chance to speak on why they believe the task would take whatever value they picked, and revoted, continuing this cycle until everyone agreed. We also improved our professional email style as a team, ensuring each of our communications was complete and respectful, with all necessary details to ensure success.

Our most used organizational tool during this sprint was a physical priority board, which was placed in the office where everyone could see it. During planning, we listed all tasks that were or could be a priority for this sprint and listed them by priority in the first column of the three-column board. As the sprint began, each person or team took on a task, signing their name to that task and moving it to the board’s middle section. As the tasks were completed, members moved their tasks into the third column at the next daily scrum meeting when reviewing their progress. By the end, the team and I could see our progress reflected on the priority board and use it to have a really comprehensive discussion during the sprint retrospective. Though the board was a useful tool in itself, weaving its use with Agile methodology made the tool an integral part of all day-to-day and planning meetings.

This project would have been more difficult had we used a waterfall approach, and the main reasons are code modularization and prioritization. These factors make it incredibly likely that the change in expectations received midway through the sprint would have set the development team back to the very beginning of their project, elongating the time it took to get the project to the tester and back to the client. It would also be nearly impossible to accomplish the projects we covered in this sprint with more than one code unless the plan is unreasonably detailed. Agile methodology makes splitting a big project into tasks and assigning them to the people most qualified much easier. One con I noticed is that people tend to begin the process with resistance; unsurprising, considering until Agile’s implementation the developer job was mostly a lonesome one, where coders didn’t have to worry so much about soft skills. In this way, Agile methodology is really changing the game. My final opinion is that in most complex projects with variables, Agile methodology is the most effective way for a team to behave and communicate to complete the project with maximum efficiency. This project in particular was no different; using Agile methodology saved us time, effort, and enabled us to create an even better product than we would have created otherwise.