Andrew Messier

Software Development Life Cycle

CS-250-T5417

06-08-2021

Final Project: Sprint Review and Retrospective

Throughout the entirety of this project, I learned a lot about the software development life cycle, the roles within, and how each contributes to the overall project. Throughout this course we explored a pseudo agile scenario through the development of a traveling website for SNHU Travel. For each week, we explored different facets of the agile development process, to better grasp how it would work within a real world scenario. For this essay, I will be taking on the role of the scrum master, and answering the following questions in the next couple of paragraphs; what are the various roles on a scrum-agile team, what are the phases of the scrum development process and how it would be different if a team used a different approach, and finally I will be explaining when it would be beneficial for a team to use agile or waterfall method for a project.

There are several important roles within the software development life cycle when it comes to agile methodology, but here are the most common roles. A scrum master is the “referee” of the agile development process. It’s their job to ensure that all other members of the team play by the rules of scrum. Whenever the rules are broken however, is when the scrum master is able to step in, and help pull the team back into a more agile friendly state of mind. The product owner is a

liaison between the stakeholders (the people commissioning project) and the development team. It's the product owner's job to ensure that tasks are prioritized based on the stakeholder's priorities and to ensure that the best possible product is being developed. It's important to note that the product owner can't really tell the team what to do, or who should be doing what. Rather this splitting up a sprint's tasks is actually left in the hands of the development team. Speaking of the development team, it's their responsibility to build whatever product the stakeholders want. Unlike other software development frameworks, the development team holds a lot of power when it comes to self-governing themselves. While the product owner can prioritize tasks, it's up to the development team to ensure who is working on what and when. This is important to note, because it allows the team to dynamically approach and overcome issues during the development life cycle.

Agile methodology has a variety of phases all of which revolves around the chaining of what is called "sprints". Sprints start with the sprint planning. In this phase, the team brainstorms ideas as well as comes up with tasks or impediments in the form of user stories. Once finished, work on the project can finally begin. The next phase is the daily sprint meeting. During this short meeting, each member of the development explains what they did yesterday, what they're going to do today, and if there are any impediments that they're currently facing. At certain points during the sprint, the team is met with a sprint review. It's during this time that the build is reviewed, so progress can be seen as well as it allows ample feedback for the team. It also serves as a marker in the case that something needs to change during the middle of a sprint. Towards the end of a sprint there's a sprint review. This is the big critique, where the build is displayed to the best of its ability. It's important to note that the build must be playable. Showing is much better

than just talking about the product and in agile is prioritized over just talking about the build. Once the review is completed, the sprint retrospective which is basically a more indepth daily scrum in which plans for the next sprint are discussed, and a new sprint goal is developed. From here the process repeats until the product is completed.

In the world of software development, there are two distinctly popular methods to create software: waterfall and agile.Waterfall is similar to the concept of building a house. First you create the blueprints, and then build the house exactly as the plans entail. The main con with waterfall, is dev teams spend a lot of time planning, instead of actually working on the project. Waterfall is great when deliverables are straight forward, and for the most part everything is understood. However, if a development team needs to create something new and innovative, agile is a much better approach. Agile is dynamic. Where a team starts with a direction, and adapts to changes or issues as they come up in development to compete for deliverables.

To conclude, the software development life cycle can be a difficult and complex process. Luckily, a development team has several methods in order to break down projects into a more easily digestible process. I think that it is important to note while this class focused heavily on agile, each development team has its own unique methods in tackling software. I expect as a dev team member on a future software project, I will see a fusion of agile and waterfall methodologies or a more refined process of one or the other. In the end I appreciate learning these different approaches and how they relate to my larger goal of becoming a software developer.