Cody Gregory

CS-250-T4471

Sprint Review and Retrospective

Throughout work on SNHU Travel the agile methodology has proven itself as a true method for success. By employing a scrum-agile approach to the software development life cycle the effects were numerous. Greater productivity, adaptability, communication, organizational tools and principles that provided a framework for successful project completion.

A large part of what makes the agile methodology work so well is the clear delineation of responsibilities that are divided into three distinct roles, the product owner, scrum master, and development team.

As the product owner you are the captain of the ship so to speak. You “steer” the rest of the team in the right direction by managing the product backlog. You reach out to all of the parties involved, namely the client(s) and end-users to refine this backlog allowing the rest of the team to focus on the tasks at hand and proceed for the most efficient development process possible.

As the scrum master you are the seasoned veteran. While the product owner manages, the scrum master leads. You are calm, cool, and collected. Your role is to be what is needed, that could be a conflict resolutionist, a mentor, or even simply another developer working on completing sprints.

As a developer you are the beating heart. Without you the product doesn’t happen, it stays a good idea in someone’s head. It is your responsibility to take items from the product backlog and turn them into something that works.

In terms of the software development lifecycle the agile approach also provides structure that helps to take a daunting task and make it more manageable such as not only identifying user stories but their end point as well.

Perhaps the biggest boon of agile however is when an inevitable interruption or change of direction occurs. Unlike the all too common waterfall strategy of attempting to plan for everything, and then desperately attempting to make any deviations follow the previously laid out directive, agile embraces change. By design agile is an iterative process, what this means is that unlike more conventional means of development, you are not waiting until the end to see how the product turns out, instead you are constantly engaging with product functionality in tandem with the evolving nature of development, leaving no mystery as to how the end product will come together.

While these aforementioned facets of agile are indeed the heart of what makes the methodology tick, the crux of all matters comes down to communication. From personal experience one of the biggest issues that prevents good communication is the lack of knowing what to communicate in the first place. Perhaps in a waterfall environment a plan is in place and it would seem redundant if not foolish to question something that is laid out so plainly, which could cause a misunderstanding or worse further down the pipeline. In an agile environment however this notion is dispatched at the core, the idea of failure is not something to shy away from but to face head first. By having such an open policy with failure, not only does it facilitate a positive environment where good communication is easy to come by, but it also has side effects such as the increased chance of innovation as well.

Additionally, one of the most difficult aspects of solving any complicated problem is the ability to break it down into simpler more manageable steps. This is where agile’s organizational tools and principles shine. Take test-driven development (TDD) for example. This is the process

of converting software requirements into test cases. One of the biggest hurdles a development team faces in development is knowing when a task is complete, as the saying goes, “perfection is the enemy of progress”. This is not to say that we do not want to provide the highest quality product possible, but it is easy to lose the forest for the trees, and by having tests that act as a goalline for sprints this creates a highly efficient process for development.

Thus in retrospect it is clear that the scrum agile approach was the most efficacious methodology for SNHU Travel. Initially user stories were created and defined, the application was to navigate vacation bookings, with features such as a filter system for vacation type, or the ability to sort by the most popular destinations. These requests were then converted into user stories where the ideas were conceptualized a step further from an idea into an actionable task. Where the scrum-agile approach shined was when the requirements shifted. We went from a wide-ranged vacation booking application to a niche wellness application. What this required was a reworking of algorithms and the user-interface. In a waterfall environment for instance this would be a devastating blow, as any pre-conceived plans of development are now obsolete. You have to go back to the beginning and start again, costing precious development time. By adhering to the agile principles however this was just another day on the job. The product owner defines the new vision for the application, the product backlog is adjusted accordingly, sprints are updated and the project iterates on itself yet again towards the new direction.