Patrick Marshall

10/15/2023

Final Project – Sprint Review/Retrospective

In our venture into using Agile processes with SNHU Travel, our team has come to find its use cases. It is a system built for flexibility, to organically tackle the day-to-day challenges of projects and surmount them without a lot of extra work. This is accomplished starting at the individual level – the Team and the roles within. The Product Manager gathers requirements and makes the backlog, communicating with the rest of the team to clear up confusion with the goal and distribute any changes to the backlog. The Scrum Master keeps the daily stand ups going, fixes any problems the team encounters, and generally keeps their focus on the team’s efficiency. The Tester and the Developer make up the rest of the team, making test cases and implementing them retrospectively.

The way these roles worked out can be best demonstrated with the process of gathering and implementing the User’s desired feature of being served destinations based on their previous travels/interests. The Product Manager gathered these requests, turned them into stories, and sent them to the team. The Tester then turned these into tests, which the developer then implemented. Each step of the process was clear, and each member of the team was quick to handle their side of the project. Problems that arose could be brought up during the next scrum meeting, any roles were assigned to those that could handle them.

The positives of Agile came about when the goals changed – originally the developer was implementing the tests in the form of a single-page list. This would then change as the desired form would be in a slide format instead. This was quickly brought up in a meeting, and any clarifications were handled by a few emails. No replanning necessary, no weeklong meetings needed.

This ability to work with one another was especially helpful in this project as there were times when the users’ desires weren’t 100% clear the first go-around. This is to be expected as we’re not perfect, but if a system isn’t prepared for changes to the plan, then inefficiencies can cost time and money. In this case, when it wasn’t clear what form the app would deliver its destination list, the way to handle this was quick. Even beyond the scrum meeting where these issues could be quickly brought up in a few minutes during the Developers turn to speak, the open channels between all members of the team, in this case the Developer, Tester, and Product Manager allowed for a quick clarification. What might have required scheduling meetings with packed schedules under a more rigid and disconnected system was rectified easily with the more open and adaptable system of Scrum/Agile.

On the theme of communication, there are various tools within the Agile ecosystem that take this aspect and push it further. While many aspects of the system tie together, the ones that most stood out to me as helping with communication were the Poker Method for planning out the project, and the project collaboration boards like Jira. The poker method, while mainly a tool of organizing and dividing product backlogs, is also a useful tool to get the team to communicate. Members had to discuss with one-another to justify their stance on the difficulty for some of the contested issues, letting them get on the same page from the get-go. The planning boards like Jira took this further, letting the backlog have a visible central location, and opening a kind of forum where each problem being worked on could be given a summary of its implementation and potential problems, letting the whole team keep up with one another between Scrum Meetings.

The Pros and Cons of using the Agile system over other systems comes down to the same thing – flexibility. While usually this is a good aspect to rely on, as you never know when something can go wrong, if you *are* certain or don’t want the distraction of other ideas getting in the way of getting something out the door, Waterfall methodology could be more suited. Of course, we don’t need to stick with one or the other and can mix and match the methodology as seen fit. Perhaps if we’re looking to get a project done that’s mostly like something we’ve done before and little deviation is necessary, we can plan and plug away at the implementation with little need for Agile adaptability. For most other cases, where the end goal isn’t %100 locked down, or if there is an expectation of the field to change tech or methodology-wise, the adaptability of Agile is invaluable.