CS-250 Final Project

Tanner W. Holbrook

Southern New Hampshire University

CS-250 Final Project

The SNHU Travel project was certainly no easy feat, especially because the client changed visions halfway through the development. However, by having a product owner who was quick to update the product backlog, gather and creating accurate user stories, the rest of the team could finish the product on time by having the information they needed.

For example, the developers immediately started working on this new vision once the Product Owner updated the backlog and user stories. As a result, the testers could write accurate and defined test cases for this new software version. Then, the scrum master made sure to keep the clients updated on progress and relayed any new information he gathered to the whole team.

User stories in agile are a big part of the scrum process and contribute immensely to the project's success. For example, it all starts with the product owner, the primary communicator between the users, clients, and the development team. Therefore, the product owner is responsible for listening to what the user wants to see in the program and asking clarifying and follow-up questions so that the information gathered is as detailed and accurate as possible for the development and test teams. Furthermore, these user stories are essentially small development jobs that get completed during the sprints. Hence, the developers will use them as a general guideline for what the user wants to see created in the project. In addition to the developers, the user stories also affect the testers. For example, the testers are responsible for creating test cases that are accurately defined, repeatable and cover the basics of the user stories to ensure no bugs are present and that the software is working as intended.

Having a scrum-agile methodology allows the scrum team to be highly adaptable and flexible when faced with problems. For example, when the team was working on the SNHU Travel project, the clients made an abrupt and sudden change in the direction they wanted the product to go. However, the Scrum Master was very efficient and quick when relaying this information to the team, and the Product Owner was able to start working on updating the backlog immediately. In addition, since the product owner was quick to update the backlog, they enabled the development team to start working on this redesign and allowed the testers to design new test cases since the user stories were changing. Therefore, the agile team could complete the project on time since they quickly react and adapt to the situation. Unfortunately, I fear that if they were using a waterfall-based approach, the whole project would have failed because they would not have adapted to the sudden change in direction. At the very least, they definitely would not have been able to present a completed project on time.

The main strategy for communication that I use is to encourage face-to-face communication between the team. Having face-to-face interactions with team members will help create a working relationship, transparency, and collaboration throughout the team. For example, suppose a junior developer goes up to a senior developer with a problem. In that case, the senior developer will be more likely to help the junior developer right then and there when compared to if the junior developer had sent an email. This face-to-face communication approach will also help increase efficiency because the person will not have to wait for the recipient to see the email and respond. Instead, they can meet face-to-face and talk, saving time and promoting a better response. However, that is not to say that an email does not have its place in the workplace. For example, take the junior and senior developer scenario once more. Suppose the senior developer could not answer the junior developer's question and instead told him he would research it and get back to him. In that case, the junior developer could send a follow-up email a few days later reminding the senior developer of the interaction and to see if he has found a solution.

The agile manifesto principles were all a big help when working on the SNHU Travel Project. However, there were only a few that stood out and drove the team towards success. For example, one principle that helped the team to be successful was "Welcome changing requirements, even late in development." (Cobb, 2015, p. 24) This agile principle helped to encourage the team to welcome the sudden change the clients wanted to see in their end product and encouraged them to work together and try their best to deliver the software on time. In addition to this principle, there are two more that worked well with the scrum events. These are, "Business people and developers must work together daily throughout the project" and "The most efficient and effective method of conveying information to and within a porject team is face-to-face conversation"(Cobb, 2015, p. 24). With that said, all four Scrum events are based on communication, collaboration, and improving the team to work better together. For instance, take the daily scrum event. This event consists of the whole team and lasts for fifteen minutes or less every day during the sprint. During this event, the team communicates what progress they have made, what they plan on doing today, and if anything impedes their work. Also, this daily meeting is an excellent opportunity for team members to collaborate on complex tasks or ask for help and advice on how to approach a problem they are facing. In addition to this scrum event, there is also Sprint Retrospective. This event consists of the whole team and primarily revolves around discussing what went right during the sprint and any areas for improvement they can make as a team. Then, they can make plans on improving processes, tools, expectations, and work relationships.

While working on the SNHU Travel project, it is apparent that a Scrum-agile based approach has numerous benefits. For example, a vital component of a Scrum-agile approach is early working demonstrations of that software to ensure everything is working smoothly, efficiently and to show the client how it is coming together. In comparison, a waterfall-based approach does not do any quality assurance or testing until the end of the project, which can be potentially devastating to the project if something significant is found. In addition to this, since agile continuously presents demonstrations of the project, they can implement changes in the project direction while also maintaining the project's scope. Again, this is significant compared with a waterfall-based approach because it is tough to change the client's needs with a waterfall project already set in motion. This is because waterfall uses a strict, detailed blueprint, so straying from this can be difficult, costly, and time-consuming. Furthermore, another advantage of agile is user stories. For example, a good product owner can capture the vision and ideas of the client and its users in the form of user stories. Then the developers can take these user stories and transform and incorporate them into the software. This function can enhance the client and user relationship by showing users that they listen and incorporate their feedback. However, agile does have one significant con, in my opinion. This problem is that agile requires a very consistent team. Since Scrum-agile teams are usually relatively small, a weak link in any of the roles can devastate the whole team. It can potentially break the entire system, which will inhibit the team from delivering the software project.

Regarding the best approach for the SNHU Travel development project, I would certainly have to pick the Scrum-agile approach. The reason for this is that the client was very broad in their description of what they wanted, which reappeared later in the project's development when they wanted to change the project's direction. Fortunately, agile-based projects are very adaptable and flexible in requirements, so this vision change was not that big of a deal. However, suppose the team used a waterfall-based approach. In that case, I fear that the team would not have been able to deliver a working project by the deadline because waterfall is all about following a strict schedule, plan and does not like adapting to change or new requirements. For this reason, I believe agile is and was the best approach for this project.

Cobb, C. G. (2015). *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach* (1st ed.). Wiley.