Retrospective

Southern New Hampshire University

December 22th, 2019

Ryan A. Manchanthasouk

Retrospective

The various roles I experienced throughout the course were product manager, scrum master, and team member.

As the product manager, found that it was my responsibility to ensure that the users/customers are satisfied with the product. To do this, I collected user stories, which provided a solid background of what the application’s features should contain. An important piece was ensuring that the Product Backlog items were clearly expressed, especially when we had a huge change in the middle of the sprint.

As scrum master, I found I worked for both the product owner and the development team. I ensured that the product owner arranged the backlog for maximum value, while helping the team understand the backlog items. Additionally, I coached the team and facilitated meetings (daily standups, sprint reviews, and sprint planning) ensuring that everyone was on task and maximized value created from the meetings.

As a team member that focused on development, it was very important to ensure that communication was clear and concise with the rest of the development team, primarily to make sure that the product developed would function correctly. As a developer, user stories that were very specific in what was needed were the most helpful during development.

In a waterfall development approach, something that may have caused a huge issue in comparison to the agile method was when the team had a sudden change right in the middle of the sprint. In waterfall, a team cannot move forward without finishing one thing at a time; it would be very difficult to backtrack and make big changes in the middle of a waterfall methodology. Additionally, in waterfall, we wouldn’t be able to test the product until the end of the full cycle, whereas in agile we would test subsequently; if we used waterfall, this could’ve caused many issues, especially with having changes right in the middle of development.

When looking to do a waterfall method, something to consider would be project size. In smaller projects, waterfall may be efficient in the sense that it is easy to manage, delivers faster delivery of the project, all processes and results are documented, and is an adaptable method for shifting teams. On the contrary, agile would not be very effective in a small project, as the development lifecycle is split off in to sprints and follows an incremental approach.

In a larger project, however, agile would be extremely effective; agile is extremely flexible, follows an incremental approach, allows team members to be self-organized, testing is consistently done, and a test plan is reviewed after each sprint. Overall, if the product owner is clear about what the user stories are, agile is much more efficient for larger projects. If a development team used the waterfall method for a large project, the team could be looking at expensive fees and high chances of bugs due to testing so late in the process.

The organization tools that were used were information radiators, face-to-face communications, and daily standups. These tools allowed the team to create openness and transparency within the team, which further maximized development. The agile principles that were practiced were “Simplicity – the art of maximizing the amount of work not done – essential,” “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software,” “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage,” and “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

The information radiators displayed critical team information such as what critical elements were being worked on, finished, or just started. Face-to-face communications allowed the team to build camaraderie because of how closely-knit everyone was. This was the most effective in the sense that it allowed team members to address issues and problems in person, rather than pushing out emails without ever knowing if the team member was properly contacted. Standups allowed the team to eliminate long and enduring meetings, saving precious work hours and minutes. At standups, team members would be brief about what would be done within the day, finishing the meeting in 15 minutes or less.

In one instance of communication, during the sprint planning phase, the team illustrated and explained each of their roles. The product owner issued out user stories, while the scrum master facilitated the meeting, ensuring maximum value output. The team members were able to communicate which user stories and responsibilities they would be able to handle.

This instance was effective in its context because it allowed team members to participate and have a sense of belonging. Team members in addition to the product owner and scrum master also have a say in how certain things should be done in the sprint planning phase. This encouraged full participation and motivates the team.

The agile software development methodology used in the project consisted of daily standups, sprint reviews, sprint planning, and lastly sprint retrospective.

Daily standups were utilized to help the team communicate what everyone would be working on throughout the day, ensuring that everyone is on the same page. Specifically, everyone on the team answered each of these questions; what did was accomplished yesterday, what will be accomplished today, and what obstacles are in the way?

The sprint review was used to demo the finished project to the product owner for their approval. This process allows the product owner to give the development team feedback and input on the project.

Sprint planning was the most important methodology, where the team and the product owner would come to agree on an overall realistic goal. Additionally, the team would define the tasks required to implement user stories and how to allocate the stories for each team member.

Sprint retrospective was essential because the team would review the entirety of the work done in the sprint. This allows the team to strategize and plan for the next sprint by implementing things that worked or disregarding things that did not work.

The agile process used helped each of the user stories come to completion because it allowed the team to prioritize user stories through the product backlog. This also allowed the team to be more organized and realistic with their goals and what could be finished within the allotted work hours.

Agile methodology supported project completion when the project was interrupted and changed direction because of the flexible nature of the methodology. The product owner was able to re-organize the backlog while the team re-planned and applied their selves to the new user stories. Additionally, because testing happens so frequently, the team could verify a shippable product.

Daily standups, face-to-face communications, and information radiators all played key parts in each of the different phases. Daily standups allowed all the team members to communicate what was finished or needs to be finished within the day, while face-to-face communications allowed the team to work closely together during each phase. The information radiators allowed the team to keep track of what was going on within the project in each phase, such as when the customer requested changes in the middle of the sprint. The information radiators also promoted customer collaboration and helped build trust and partnership between the development organization and business owners.

“Simplicity – the art of maximizing the amount of work not done – essential,” enforced the team to keep things simple and uncomplicated, to easily move forward with each phase. “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software,” pushed the team to work around the changes in the middle of the sprint and ensure that realistic goals were complete. “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage,” ensured that the team understood the importance of flexibility in each phase. “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly,” set the standard on the iterative nature of agile, allowing the team to understand the process.

During the sprint review, the team got feedback from the product owner on how the current state was looking. All the phases prior, such as the sprint review and daily standups, led up to this phase, where the team gets together to ensure that the current product is meeting the customer’s standards. This phase was important because it allowed the team members to collaborate and find out what precisely needed to be done. The product owner expressed the sudden changes to the project and the team was able to work out a strategy on how to get the product shippable.

Before my experiences in this class, the only role I saw myself fulfilling was as a developer. I thought I would only be writing and testing code. After taking this class, my previous view has changed from being just a developer or tester to working as a scrum master. I would enjoy facilitating and supporting the product development, while coaching and helping team members out.

A tool that I would like to better understand is Jira Software, which allows a scrum master and product owner to plan, track, and manage all agile software development projects. Additionally, I would further like to experience use of information radiators and how they are used in real-time. I would like to experience or observe how “Simplicity – the art of maximizing the amount of work not done – essential,” might impact a project team that is newly implementing agile scrum.

During the retrospective, the team members went over processes that had a positive impact on the development team. The team also went over obstacles, such as a huge change in the middle of development and how it was difficult and stressful for the team to overcome it. In this communication, I believe that the scrum master best exemplifies the role I would like to have in the industry. As a scrum master, I would facilitate the retrospective and ensure that the team is being productive and impactful, making sure that we have a good use of time. I would be completely comfortable with surrendering full control over to the Product owner and team and enforce self-organization. I think that I would have a positive impact by coaching and assisting when team members are in a clinch. Above all, I enjoy working with the functionality of things, which is one of the scrum master’s main roles; to ensure that there is cross-team functionality between the users and the team.

Moving forward, my personal and professional goals are to attend and pass a Microsoft Software and Systems Academy course on cloud application development. This course would allow me to pursue more schooling and certifications, along with providing me an entry-level job into the software development industry. Another goal is to be a certified scrum master. To achieve this goal, I am seeking to study more agile scrum methodology books and attend a scrum master course. Additionally, I want to become a better developer in general. To do this, I have outlined a personal project with the purpose of constructing an application for the fitness industry that would link up personal trainers or coaches with people that have specific goals.