**Final Project: Retrospective**

**Rylan Champion**

**CS 250 Software Development Lifecycle**

**DEC. 7. 2023**

**Abstract**

In this paper, I'm looking back at the past seven weeks, wrapping up my "project." I'll share what worked well and what didn't based on the roles and experiences I had. We'll talk about the advantages of using agile methods and how teaming up with others in different development roles makes a project better. The conclusion touches on the organizational tools and principles I want to get better at in my future work.

**Sprint Review and Retrospective**

With this course I have studied and implemented the Agile method as well as partaken in an example project for a company named ChadaTech, a company that specializes in custom software design and development for both domestic and international clients. The prompt that we followed was that “for many years, the company has been developing their software using a waterfall development model. Recently, ChadaTech has considered shifting all of its development teams to an agile methodology that uses the Scrum framework. They think the agile approach will enhance their products and build a more cohesive corporate culture. However, before the entire organization makes this shift, ChadaTech decided to pilot this process with one small team: yours. Your team has been assigned to develop an application for SNHU Travel, a travel agency seeking to expand their client base with new tools for their customers. Your team will take a Scrum-agile approach to developing the application.”

Having participated in each agile role, Product Owner, Scrum Master, Developer, and tester, I will now finish up this course as the Scrum leader again and present a retrospective and review about how my team and I completed tasks and what we learned.

**Product owner:**

To start Demonstrating how the various roles on my Scrum-agile Team specifically contributed to the success of the SNHU Travel project. I will begin with the product owner. In this hypothetical with me as the product owner, I played a crucial role as the bridge between the development team and the stakeholders. My primary responsibility is to ensure that there's a seamless flow of communication between these two parties. To achieve this, I need to be diligent in gathering the right information regarding the needs and desires of both the users and stakeholders.

In this regard, conducting meetings and utilizing various forms can prove to be invaluable tools for collecting and analyzing the necessary information. User stories, in particular, are a valuable asset for the scrum team. They essentially serve as the blueprints that outline what needs to be accomplished during each sprint. In this course I partook in a meeting with some customers of SNHU Travel and they spoke their minds about possible applications that could be implemented into the program and I took it upon my self to translate these into user stories that then the developers have tasks in order from high priority to low to finish during each sprint of the agile process. For example,

|  |  |  |
| --- | --- | --- |
| **User Story Number:** | 3 |  |
| **User Story Name:** | Edit Profile settings with vacation options |  |
| **User Story Size:** | large |  |
|  |  |  |
| **User Story Value Statement:** | As an end user I want to be able to edit my profile setting so that I can be recommended vacations based on my preferences. |  |
|  |  |  |
| **Acceptance Criteria:** | 1.Click "profile" from a menu where I will be able to customize my settings.  2.Choose desired locations on a Wishlist. 3.have multiple preferences to fill out such as:   • Address  • about you  • privacy and security  • contact information  • payment information |  |

Utilizing tools like meetings and user stories empowers the product leader to construct a backlog and visually comprehend the critical components necessary for realizing the client's vision. These tools facilitate the efficient prioritization of tasks and delegation of work. As I crafted my own product backlog and user stories, it became evident that visualizing the workload in this manner significantly enhances project organization and expedites its progression

**Scrum Leader:**

Taking on the role of a Scrum Master comes with its share of benefits. I found it appealing that there's a strong emphasis on teamwork, with someone on the team genuinely pushing for everyone's success, sometimes even more than their own. Handling key development aspects like sprint planning, daily standups, backlog grooming, and sprint retrospective meetings is part of the Scrum Master's job description. These gatherings play a crucial role in promoting teamwork, efficiency, accountability, focus, and self-organization.

In the context of sprint planning and daily standups, it's essential to hash out what got done, set goals for the day, and assign tasks. Having daily goals that everyone on the team can agree on makes it easier for individuals to tackle their work and avoids getting stuck on one task for too long. These meetings also provide a space to spot and resolve conflicts, trim down unnecessary tasks, and tweak estimates based on new information.

During my sprint as a Scrum Master, creating an agenda for the daily stand-up meeting was one of my tasks. Learning how to use this tool effectively made me realize the importance of these meetings in kicking off a sprint and starting the workday on a positive note. The seamless coordination of these Scrum ceremonies ensures that nothing slips through the cracks and that everyone on the team is responsible for their work. Communication is vital for every member of the development team, and the Scrum Master plays a key role in making sure information flows smoothly among team members, and that everyone is clear on their goals and tasks.

**Tester:**

A tester within an agile team plays a pivotal role in formulating and executing tests to ascertain whether the product meets or falls short of expectations. Through frequent collaboration with the product owner and the broader development team, this ensures a complete understanding of the criteria that user stories must fulfill. Testing is an integral part of the design process, and the tester bears the responsibility of discerning which aspects of the project are functioning correctly and which are not. Importantly, I’ve come to understand that setbacks are not entirely detrimental. While a tester may encounter occasional failures, these instances are swiftly addressed, and the insights gained are openly shared with the entire team.

In the broader context, user stories prove immensely beneficial in constructing test cases. Specifically, the "acceptance criteria" stands out as the primary source of invaluable information. Additionally, considering the size and priority of each story proves advantageous for establishing a shared understanding of how to distribute the workload among team members.

Upon receiving user stories from the product owner, there may arise a need for supplementary information. In such instances, the tester may initiate communication with the product owner to seek clarification before embarking on the development of test cases. In my capacity as the tester, there was an occurrence where additional details were required from the product owner. I consequently drafted the following email to address the situation:

Dear Product Owner,

I hope this message finds you well. As I begin working on the test cases for the user stories you provided, I require some clarification on certain points to ensure accuracy. Specifically, for User Story #2:

What types of filters should go in the personal profile section? Should this be a checkbox format?

Are users allowed to specify their price range? If nothing is selected, will all vacations of any budget be shown?

As well as the “hot deals” function for the program. To facilitate a smoother development process, I kindly request additional details on the specific expectations for the “Hot Deals” feature. Are we focusing on random, cost-effective vacations, or is there a particular aspect like a loyalty program that should be emphasized?

Your insights on these aspects will greatly assist in developing precise test cases.

Thank you for your prompt attention to this matter.

Best regards,

Rylan Champion

Tester, SNHU Travel

Then as well as emailing back and forth the project can sometimes take a left turn and have completely different requirements. When this happens, the Product Owner will need to re-prioritize the Product Backlog so that the team can focus on the features deemed most important. Typically, the new features would be broken down into user stories that are deemed “ready to work” by the team. Then the developer takes hold.

**Developer:**

Developers in an agile team play a crucial role that sets them apart from those in a more structured waterfall model. Unlike in a plan-driven approach where developers may work alone on coding, in agile, their tasks include various aspects like software design and testing. They actively talk to users, testers, and product owners to refine project requirements and share the responsibility for the software's success. Good communication within the team is important, and developers need to stay aware of any changes during the development process, quickly updating their software as needed.

In my role as a developer, I faced challenges when changes were made during code development. Agile teams, however, handle these situations well through clear communication, potentially ensuring that deadlines are met despite modifications. Changes affect every stage, from the product backlog to user stories and test cases. At the same time, developers continue coding tasks, adjusting as necessary. Seeking clarification from testers or product owners during code changes was crucial, adopting a specific and detailed approach by asking questions about functionality and desired outcomes to make the workflow smoother.

The flexibility in agile methodology lets developers adjust their approach. Expecting changes or updates is a key part of agile, with room allowed for modifications in features or other project aspects. User stories, initially seen as placeholders for conversation, require developer interaction with product owners, testers, and users, especially when significant changes alter project goals.

Discussions often arise about the merits of agile versus waterfall methodology. The idea that these methods are mutually exclusive oversimplifies the range of adaptive and plan-driven approaches. My experience showed that the roles and strategies I used would have been different under a waterfall method. It's essential to recognize that neither method is inherently better; each has its strengths based on the project's characteristics. Agile is beneficial when changes are expected, the final product isn't clearly defined, or quick delivery of working software is crucial. On the other hand, the waterfall method is recommended for projects with well-known requirements, predictability, and minimal expectations of changes during the project's duration.

**The Agile Method Looking back**

When beginning this class I had no idea what to expect in a software development job and was pleasantly surprised to see how modern and straightforward the agile mindset is. I also saw how necessary it was to complete this project. The meetings most of all were something I began thinking was possibly a waste of time, but learning what they are helped me understand how necessary they are:

The daily stand-up meeting begins every workday morning, with a brief duration of no more than 15 minutes. Although the atmosphere during these sessions is generally light and enjoyable, the primary aim remains informational. The purpose of these meetings is to provide each team member with an opportunity to share their achievements and outline their tasks for the day. Specifically, individuals address three key questions: "What did I complete yesterday?", "What will I work on today?", and "What challenges my work?". The focal point of the meeting is to encourage team members to take ownership of their tasks, fostering accountability and ensuring everyone is informed about each other's progress. These daily interactions significantly contribute to team collaboration, efficiency, responsibility, focus, and overall organizational effectiveness.

As for product owner, scrum master I didn’t even know those roles existed, I thought it was just the manager and developers and testers. While the agile approach is not without its challenges, such as managing evolving requirements and striking the right balance between rigidity and flexibility, its strengths in promoting adaptability, accountability, and continuous improvement are evident. In essence, the Scrum-agile methodology has proven to be an effective and transformative approach for the SNHU Travel project, instilling in me a newfound appreciation for the agility needed in the ever-evolving landscape of software development. In conclusion I believe that the agile method is far more helpful in terms of the SNHU travel project than waterfall would have been and I now understand what is required of me when I one day join a software development team.