CS250 Final Project:

Alexander Dollison

SNHU (Southern New Hampshire University)

08/12/2023

**Abstract**

The last 7 weeks of this class has taught me many things about agile development, the work that goes into it, and how much each role relies on one another. Learning about what works for agile and what does not, I will discuss the benefits of the agile methodology and how the various team members can contribute to a project in ways that help the product produced succeed. I will conclude the paper by discussing organizational tools and principles that I want to gain more experience with as I enter my profession in the future.

**Retrospective**

Throughout the course, I was given the opportunity to take on several distinct roles of the scrum team, including as a Scrum Master, a Product Owner, a Tester, and finally a Developer. All these different perspectives allowed me to understand the full picture through individual roles. Each role relies on the rest to be able to do their jobs effectively, especially when learning a new methodology like Agile. Communication is key, and each member must take responsibility for their roles for a product to be successful.

First, as a Scrum Master, collaboration and planning were my key responsibilities. One of the most important roles for an organization, the Scrum Master plans sprints and daily standups, grooms the backlog, and leads the sprint retrospective meetings. Each of these “ceremonies” are extremely important as they assist with collaboration, efficiency, accountability, focus, and organization.

Sprint planning and daily stand ups discuss what needs to be accomplished during that day and allow team members to decide who needs to do what during that day. Having these goals ensures that everyone is on the same page and work is not getting done twice, or that parts are not being missed. They also assist the team in identifying and resolving areas of conflict, removing unnecessary tasks and correcting estimates using newly discovered information.

As a Scrum Master, I was required to create an agenda for the daily stand-up meeting, which taught me how to utilize my time and that of my team effectively, and how important morning meetings are to the success of a product. Each ceremony offers different benefits, which when put together, offer cohesive solutions to team cooperation and product management.

The role of Product Owner allowed me to see through the eyes of the “control rod.” Product Owners are responsible for understanding the product, making decisions about the product, and managing what stories are prioritized over others. They are the intermediary between the client and the development team, and their goal is to share the unobstructed vision of the product in a way the team understands and can accomplish in the given time. It is incredibly important that a Product Owner does not micromanage the team, but instead guides them in the right direction while simultaneously allowing them the freedom to work and accomplish goals.

User stories are essential in delegating tasks and prioritizing workloads for the development team. Product backlogs and user stories are created by meeting with clients and team-members to acquire their feedback on issues or to clarify aspects of a product. When the goal of a product is clear, user stories allow the team to prioritize certain aspects of a product deemed more important over those that may be less significant or partially unnecessary. When user stories are created, the product owner creates a backlog to track what has been completed, as well as what still needs to be finished. These tools, when combined, speed up the progress of a project and allow the team to work efficiently.

The role of a Tester is to repeatedly test the coding portion of the product to find bugs, report those errors to the development team, and ensure communication (or lack there-of) between the different team members does not lead to the failure of the product. Testing is a constantly occurring process within agile development and is the responsibility of the Tester to identify what parts of the product are working and what is not. The progress and results of the testing process is shared with the rest of the team, and it is important to note that rather than run perfectly, these tests only must meet the “acceptance criteria.”

After receiving the user stories from the product owner, additional information may be required, and a tester may reach out to the product owner for clarification before starting development on the test cases. Whenever I needed more information, I sent emails requesting clarification on various aspects of the product. Below is one example.

“Dear Product Owner,

I appreciate the punctuality in submitting your user stories to me, and the detail of which you wrote them. I am working on the test cases to ensure the finished product will meet your needs and standards. To continue my work, there are a few more details I require to optimize the process and deliver a set of satisfactory test cases.

User Story Two

Where and how would you like the “Top five destinations” feature to be held? Would it be in the profile, or would it lead elsewhere in the search results page?

Are there specific settings you would like included in the larger picture of this feature?

User Story Three

Did you want to include budget settings? If so, where did you want those settings to be located?

Did you want to include all sites with deals or just the most popular ones?

User Story Five

When did you want to include the “Your Top Five Destinations” feature? From the beginning or only after the user has been to five destinations or more?

How did you want to grant these rankings? Based on a rating system or based on how popular the destinations are?

Thank you again, and I look forward to further contact,

Alexander Dollison”

This is merely one example sent to the product owner to effectively relate my needs and prompt a response from the product owner to progress the project. This encourages cooperation and collaboration amongst team members by inspiring further discussion about a project's details. As with aspects and roles of agile, communication is essential.

Developers on an agile team have the incredibly significant role of creating the code for a product and bringing the product to life. Whereas with waterfall a developer may work on a product alone, the agile methodology has them cooperating with the rest of the team just the same as any other team member. This allows implementation of new code to be as smooth and relevant as possible, making changes on the fly when necessary.

Even with major changes suddenly occurring, the agile methodology allows team members to communicate how to implement these changes in an effective manner, which could allow deadlines to be met even if entire blocks of code must be changed. Communicating with the tester means that a developer can fix bugs quickly, and meetings with the Product Owner means new aspects of the product can be implemented as they are obtained.

Discussion about agile and waterfall usually leads to arguments about the two, and "the

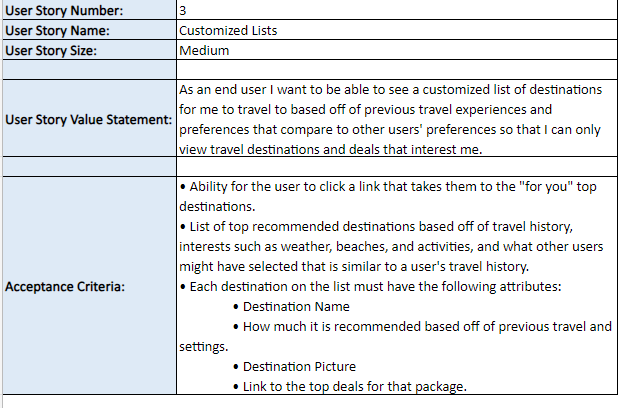
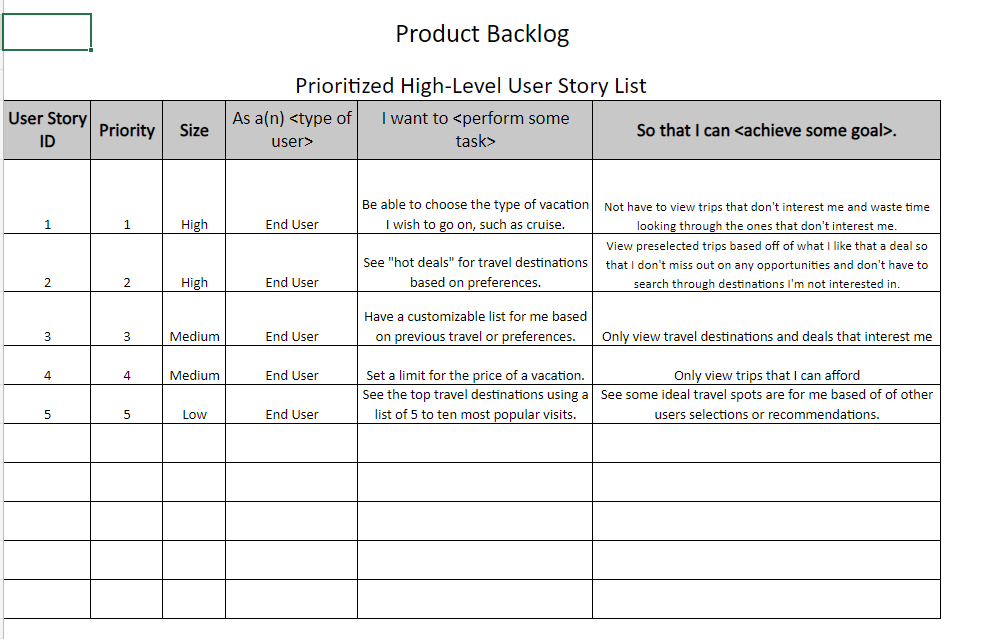
typical discussion that compares agile and waterfall as if they were two discrete, mutually exclusive, binary choices oversimplify what should be more accurately thought of as a range of adaptive and plan-driven approaches." (Cobb 13) While the approaches of waterfall and agile may be different, they both have benefits that the other simply cannot accomplish in the same way or with the same effect. Agile allows a cycle of actions to occur that ensures all aspects of a product to be complete and several stable versions of a product exist before its release, which can be extremely helpful in the case of long-term projects. Waterfall is great for short term projects that require less effort and thought, where all aspects of the project are known and discussed before work has even begun.

Each phase of the software design life cycle is just as essential as the last, and moving through each phase allowed me to experience what agile development looked like from the inside. The following breakdown from my role as Scrum Master clarified how these meetings were useful to the project overall:

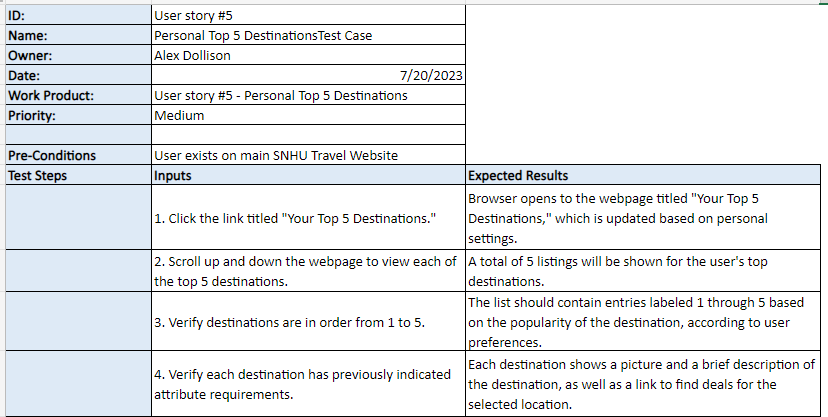
*Sprint reviews are naturally done towards the end of a sprint but can also be done before the sprint ends. If you have sprint reviews every week, it will be easy to review what has been finished as well as what still needs done and can lead to clearer expectations during a sprint. Sprint retrospectives are always done at the climax of a sprint and allow for the team to go over what they did effectively and what else might still need to be worked on. All these events are important to keep the team on track and ensure everything is done in a timely manner while keeping quality as the most crucial factor.*

It seemed tedious to me at first, having meetings every day and more every week to discuss what needs to be done, but now I realize how crucial these meetings are to the success of a product. Allowing these reviews to occur and participating in them spurs on the development of a product as everyone discusses how to progress the product.

Creating a product backlog allowed me as the product owner to break down the project from beginning to end and document user stories in an efficient manner. Below is an example of this.

 The backlog assisted me in accurately creating user stories and allows the team to start in a more realistic place. This also ensures the more important aspects of a product are covered early on and smaller tasks are delivered as soon as a few days later.

Test case creation allowed me to easily assess user stories and begin development. An example of one of my test cases is presented below.

 The ”Expected Results” category allowed me to understand how important clarity is for this step. Understanding what is expected of a step may not be strictly necessary, but it does assist developers with creating their codes efficiently.

The “product” in question hit a drastic shift during one of the modules which saw notable change in the outcome of the product. Luckily, Agile’s flexibility allowed the team to re-center and concentrate on implementing the necessary changes without much loss of time. Below is an example of another email sent to clarify some follow-up questions that I had after the change.

*To: Product owner*

*CC: Tester*

*From: Developer*

*Subject: Request for Specs*

*Good afternoon, Tester,*

*While doing my work on the user stories, there were some details which lacked clarification. For our work to continue, I will need more information regarding the requirements. Please provide the full scenario for what the requirements are for the product, including your desired menu typing. As an example, would you like the menu to be a dropdown menu or a checkbox? With the Mobile Application, would it have the same functionality as the web page, or do you want it toned down? Are there certain features you would like prioritized? All details would be appreciated and would assist in the development of the product. If you would not mind, I would also like feedback on the deliverables. Thank you and I look forward to our continued collaboration.*

*Best regards,*

*Developer.*

This is another example of interaction between team members that encourages collaboration. Developers gain clarification from their teammates so that they can accomplish their goals, but discussion allows specific tasks to be finished with a higher degree of clarity because of this communication. Waterfall would not be able to support such a sudden change as this, and agile’s flexibility allowed the team to shift without really losing momentum which means the product was able to succeed and complete mostly on schedule. With agile, each member is like a cog in one big machine. When everything is working together, the machine runs smoothly.

**Future in the Industry**

This project has allowed me better insight into my future profession as a computer scientist and software engineer. Having knowledge about agile will certainly be helpful when it comes to joining a company, as regardless of whether we do waterfall or agile, I will have at least been introduced to both concepts. While my goal is in the development of artificial intelligence, I believe agile will play a role in that subject as well. I am more comfortable than ever in my ability to adapt to the needs of the industry and look forward to seeing what it has in store for me.

If I were to send communications to a prospective employer, my goal would be to convey concisely who I am and what I want to do in the industry. I would list my relevant experiences, goals, strengths, and weaknesses. Clarity is important even in everyday life, and I would not want to start a career with lies. My fiance encouraged me to pursue a career in computer science and I have not regretted a moment of it. I will continue to learn and grow in the hopes that one day my goals will come to fruition.

References

Cobb, Charles G.

The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach. John Wiley & Sons P&T, 2015-01-26. VitalBook file.