# CS 250 Sprint Review and Retrospective

Jad Alrehaoui

Jad.alrehaoui@snhu.edu

Southern New Hampshire University

## Roles Contribution

Starting with the **Product Owner**, who is responsible of maximizing the value of the product and the work of the development team, the role contributed by being a liaison between the team and the stakeholders who in this project were the clients. Conducting interviews with the clients and asking them for feedback to optimize the value of work the development team performs. Keeping a wide perspective on the trends and technologies the product owner was able to keep the application up to date and trendy. The product owner also translated the verbal requirement provided by the users (stakeholders) into meaningful requests for the team to work on which helped the team what is really being asked for. Finally, the product owner prioritized the features to be worked on by ordering the items in the product backlog and ensured that the team understands each item by communicating with the Scrum master.

Jumping to the **Scrum Master,** who was responsible of ensuring scrum is understood and enacted. The scrum master was the key communicator and servant for the product owner and the team. They were more involved in the team and what’s happening on the team, what is blocking the development process and what is done. By conducting daily standup meetings, the scrum master allowed the team to have efficient communication, find out what’s blocking the development and help if possible, keep track of where they are and what they are planning to reach, and ensured everyone on the same track. In the example of the scrum daily meeting provided earlier, we visualized how the scrum master explained the role of the product owner to the team, which was one of their duties to pass the information coming from the product owner to the team.

Talking a lot about the **Team,** the team is a self-organized group of members, they decide how to turn the product backlog into increments of potential releases. Having a cross-functional team ensured that everyone knows something about everything. So everyone on the team can collaborate or cooperate to come up with the release.

**SDLC**

SDLC is the Software Development Lifecycle, a framework that tells us what happens at each step of the development process of an information system. It is broken down into the following steps:

**Planning:** is where we set up the goals and outcome we want the system to reach, what is feasible and what is not and considering the given budget and resources at hand.

**Analysis:** is where we gather information about the system requirement and review the requirements.

**Design:** is where we model the system based on the requirements gatherer in the previous stages, we visualize the system on paper, by modeling databases, relationships, objects of the system. How these endpoints interact, what information is needed when, what kind of access is given to certain scopes.

**Implementation:** is where we start using programming languages to build the previously designed system.

**Support:** is where we make sure the system is always in a running state, modifying some components for the better, and supporting users.

Taking this approach helped each user set up a plan of work before they jump into coding directly, they were overseeing the scopes and where they want to put more effort. This approach also helped us manage our time by prioritizing the stories so we reach our goals efficiently.

**Scrum-agile Interruption**

In an agile project, uncertainties and changes are encouraged, so as the requirements changed, we were able to go back to the early stages of planning, modified our models and stories and get back to work on the updated tasks. It was easy because we were using the agile methodology and with some leading technics we were able to continue and complete the project.

However, over looking the whole project process, we think these changes interrupted our work flow and changed some areas in the project. Saying that the project would be done faster without it being interrupted is fair, but because it is an agile project, we are sign up for uncertainties and changes so it was fairly easy to adjust and continue.

**Communication**

In this project, we were supposed to communicate by using the university’s online platform, I was assigned as the scrum master. I introduced myself as the Scrum master, and tried my best to be communicating with the product owner and the team effectively. The product owner assigned the Microsoft DevOps platform to be our communication platform, and to get familiarized with it. I made sure the team is following orders and not having any issues doing so. I also picked a motto for our team “**Work SMART not HARD”,** which I believe my fellow colleagues found interesting and motivating. I also held daily stand up meetings where I gave time to each member on the team to tell us what they’re doing and what they did yesterday and what difficulties are they going through. Communication was a key factor in this project and knowing where everyone is standing at all times was so effective to complete the project.

**Tools evaluation**

As mentioned before, Microsoft DevOps platform was the tool we used in this project for communication and task assignments. Personally I prefer Slack since it is used more widely and more people are already familiar with it. However it was a good experience getting to practice a new tool and getting familiarized with it. This tool served as a white board for the tasks where members on the team are assigned with tasks, it also served as communication platform where we could comment and communicate our insights and problems. The best feature was that this tool allowed us know exactly where everyone is on the project and what are they going to be doing later, even when we are miles away from each other, in a way it served as the work office.

This tool allowed us to be fully engaged in the process, update the process in real-time, view the status and issues and to scale the project and provide reports.

**Effectiveness Assessment of Scrum Agile**

During this project we found some pros and cons of the Scrum Agile approach that we can learn from to emphasize or to adjust for the next agile project.

Some of the PROS we found are :

* The ability to complete the project quickly and efficiently
* The team got clear visibility of through scrum meetings
* The ability to get feedback from the customer and adjust the requirements
* The individual effort was visible during daily stand ups

Some of the CONS we encountered are :

* Uncertainty and adjusting to it
* Daily meetings were sometimes frustrating, some team members felt behind while others got demotivated when they were ahead.
* Changes in the requirements got the team members demotivated at first because their workflow got interrupted

In conclusion adopting the Scrum Agile approach was a good choice, since from the beginning of the project we knew that we have some uncertainties, and we want to be fast. It also helped us visualize the effort being made without having a traditional project manager who is responsible of tracking it. It made the job easier as it provided a full visual of the process and the work done at any time. Finally it helped a lot with communication and build a team chemistry for upcoming projects.

Thanks for reading.

Jad Alrehaoui