**Success with Scrum-Agile**

Connor Corson

Southern New Hampshire University

December 08, 2021

Being a Scrum Master means more than simply being a boss. It also means being more than a creator. I am here to lead you to success. I do this by guiding you through the rough patches, especially when we learn something new. I asked that we switch a whole new way of development. The Waterfall method consisted of five stages: requirements, design, implementation, verification, testing, and development & maintenance. (Adobe 2021) There were some perks of waterfall, completion time, an end result we can all assume, and potentially money friendly. However, we found a new way of production that allows us to make errors, correct them, and get work done more efficiently!

Scrum-agile uses numerous smaller teams that work together to create a project. Like a rugby team training for a big game, Scrum encourages them to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses in hopes of improvement. (Atlassian 2021) Scrum consists of content that can help a big project become possible. With sprints, a large order of authority and development purposes, it truly is the better approach to getting our project done. To go into more detail on the role of each and every one of you, let’s break down what each team member did in the success of our project.

Let’s start with the original creator of the idea, the product owner. He is in charge of defining stories and team backlogs. Stories are a way of explaining features of our software. They are usually done by the end users, or product testers. They are significant for helping us create the best software possible. Team backlogs are considered our schedule. In order for us to keep everything going the way we want, we need to keep ourselves on a strict schedule. That is what the team backlogs are for. It is simply a list of requirements needed within each sprint. A sprint is a time period of amount of work requested. To reflect on all of this, The product owner is who created the idea of the software, who is in charge of getting our feedback, and creating our schedules to help keep us in line. They also participate in some meetings to help understand any downfalls or roadblocks in the way! A huge responsibility the PO’s have, but are very ideal in the success of our team.

Now let’s get into what my role is in this process. If you did not know, I am your Scrum Master, I like to consider myself the flexible one. I have many different responsibilities that help better our process. I am responsible for our daily scrum meetings. These meetings are vital to our success because it requires everyone to be on the same page on things, which is obviously very important. I also moderate the meeting to make sure it goes smoothly, and create solutions to each concern. When it comes to the work at hand, I am responsible to keep everyone in check, and to make sure you are all accountable for your work! I will occasionally help you with things, and give advice on the occasion.

Next, we have the developers. Developers are the builders of our process. Developers do exactly what you think they do, they do the work! Developers are in charge of efficiently getting the job done! Alongside their scrum masters, developers can thank their success to the guidance of scrum masters, their acquired knowledge, and the daily scrum meetings. All of these allow the developers to keep in check, stay focused, and being on the same page, which is very important.

Lastly, we have the testers. Testers are like the little men that test different ways to push a house down, to ensure it will stand tall. Testers were blessed with simply testing the design, the process, and the success rate of our software. Testers are very important because they are in charge of making sure the software works, and there are not errors.

To sum this all up, why do we have all of these roles, and why are they significant to each other? I believe that each role depends on each other to succeed. Without the Product Owner, this whole process would be vacant, and the Scrum master would not know where to begin. Without the Scrum Master, the developers would not be on the same page, and our Team Backlog would not be followed. Without the developers and testers, the work would simply not be done, and the whole operation would fall apart. Everyone has a role, and without being followed in unison, our operation would fall. Let’s say I work at a giant store with numerous departments: meat, produce, bakery, the floor, ect. Without the role of each department following a strict schedule and job order, the grocery store would not get enough sales and crash. We all have a role, follow it and we will succeed.

I briefly went over a user story, a way for an end user to write their take on our software. It can be used to write pros, cons, and improvements requested. These user stories are very important to help better our software! The SDLC means the Software Development Life Cycle. It consists of the project planning, analysis, design, implementation, and support. It is practically the heart of Scrum-agile. So to take a user story and make it a reality, we create a game plan and analyze it. We have the developers design, and implement said story into our software, and the testers will test it. With the help of Scrum’s methods, we can implement new things into our software with ease! I personally found this to be easy when working on making my own user story. As I worked on this assignment, I was able to think about how I would incorporate my ideas into the software, and it gave me a very vivid idea on what the user story is truly designed for, and the purpose of SDLC and its significance!

The downfall of the Waterfall method was the theory that they cannot turn backwards. Everything about that method was simply looking forward. Scrum-agile has a very different philosophy in the matter. Taking in input to incorporate it into our software is the stepping stone in our process. Let’s say that the Product Owner gives us a new task of incorporating something new into our software. We would just simply take a step back in our process, and continue using the SDLC to make things happen. We would make a plan, analyze and continue. Interruptions and interferences are just a part of the process of success. Without it, we would not learn from our mistakes and take said mistakes and turn them into success! I learned that when I wrongly estimated how long it would take to finish a homework assignment, I learned to give myself extra time to fulfill my assignment and have time to spare to study other things! Failures are the stepping stones to success, and Scrum-Agile taught me to accept failure.

Before switching to Scrum-Agile, I always thought that meetings were pointless. I was under the impression that meetings were always unorganized, toxic, and overall unhelpful. Communication was never an option, and meetings consisted of people arguing, and solutions were only a thought, but never implemented. This old meeting analogy was a reality my old job dealt with every month, nothing was truly solved, and it truly was pointless to even go to these meetings. Scrum’s method of daily scrum meetings fixes any issues with meetings, and taught me that meetings were more than unorganized gatherings. With the help of a scrum master organizing each discussion, taking notes, and creating solutions, these scrum meetings help each individual understand a problem, and mentally helping them give a result. Having a scrum master keeping things in check also helps keep the meetings short and sweet, so the team can go back to work with a clear mind and a developed understanding of the assignment at hand.

I can discuss our organizational tools all day! With Scrum having numerous ways to stay efficient and organized, it truly would be a time trying to hop off the horse! I mentioned the Team Backlog, created by the Product Owner, to keep a sophisticated schedule in check. We also have the daily scrum meetings. These meetings are a short session to discuss any roadblocks that are or could be found, and used to give solutions and to keep everyone on the same page. But do these relate to scrum events in any way? Of course they do! A scrum event is simply an event that is time based. It is an action that has a duration. These tools we use are very important in making sure that the scrum event in question is finished by the time the duration ends. If we have a project due in two weeks, and we do not know what is going on, we host scrum meetings to get everyone on task, and understanding the assignment. If we do not know the actual time duration, we have the Team Backlog to show you what you are asked of, and you know what needs to be done each day in order to succeed.

As I have mentioned numerous times, Scrum-Agile is a proficient approach to completing a big project or software. But are there cons to Scrum-Agile? As any big company would deal with, having unmotivated developers can be a downfall. Without scope creep, having workers that are not committed to the success of our company, Scrum-Agile may not work with the efficiency we need. I would also say that Scrum-Agile requires more experienced members. Throwing a newbie into the mix, there will be a number of issues. We would have to teach them our process, have them ask questions and answer them, and have to keep an eye on them constantly to ensure they are doing things correctly. This could potentially slow down our process, and have the time backlog be harder to achieve. With this being said however, I still believe that Scrum-Agile is the better process than many others.

CITATIONS:

Adobe: <https://www.workfront.com/project-management/methodologies/waterfall>

Atlassian: https://www.atlassian.com/agile/scrum