Nathan Mailhiot

Southern New Hampshire University

06/17/2022

Abstract

This paper will cover the agile development process. It will outline the different roles involved in the scrum process as well as the various phases of the SDLC. It will also outline the differences between the waterfall approach and the scrum-agile method. The waterfall approach will also be outlined and reference to which method to use in what scenario will be demonstrated as well.

Keywords:

SNHU- Southern New Hampshire University

# **Roles on Scrum-Agile Team**

There are many different roles involved in the scrum-agile process. Each role plays a key factor in the scrum-agile process succeeding. Each team member needs to embody this process for it to work correctly. Each team member must be committed and focused, communicate openly, respect one another, and have the courage to speak their mind [1]. The different roles involved are product owner, scrum master, developer, and tester. These are the typical roles in the scrum-agile process and were the ones used in the SNHU travel project.

**Product Owners** have multiple responsibilities that are critical to the scrum-agile process. The product owner is responsible for talking with the shareholders and making sure that their requirements are met. The product owner must have excellent communication with the scrum master as well [1]. The product owner must create the backlog. The backlog is what items are needed to complete the goal at hand. This must be organized in a manner that allows the product to be finished in a timely manner, as well as be broken down into more digestible goals for the members of the team. The product owner may also have to create user stories that are required for testing. Product Owners take responsibility for the product as well as the flow of the project. During the SNHU travel project as the product owner we had to put together user stories.

**Scrum Masters** are a jack of all trades, and a master of communication. The scrum master is the one who communicates for the team. Whether that be with other team members or others outside of the team that may be causing distractions, the scrum master is in constant communication. Leading the daily scrum meetings is one of the requirements for this communication to work most effectively. The scrum master is up to date on all agile processes and helps the product owner refine the backlog [1]. A genuinely great scrum master can have a team lead themselves and not be needed. They are also responsible for making sure sprint reviews and retrospectives are done.

**Developers** are the heart and soul of the scrum-agile process. They are the ones who build the vision for the product. They need to be open to communication and be self-organized. A developer needs to be an expert in problem solving, as well as estimation. The developer should not hesitate to communicate how long they estimate any given task would take them and reach out for help when necessary. The developer works from the user stories that are provided and must reach out for any clarification on the stories should there be any confusion. As the developer for the SNHU Travel project we had to rework part of the application to fit the new requirements. This meant only editing parts of the application that were outlined by the product owner.

**Testers** play a key role in the scrum agile process. They are responsible for making sure the user stories are tested. They review the story and communicate with the product owner on whether the story is clearly laid out or not. They make sure that all edge cases are thought of and that unique features are thought out. While working on the SNHU Travel project as a tester we had to go through the different user stories provided and rewrite them, so they are more flushed out. This back-and-forth communication allows the tester to create a more flushed out user story for their tests and the developers.

# **Phases of SDLC in agile approach**

There are many distinct phases of the scrum-agile development process. All of which need to work simultaneously to keep the development process in a constant state of flow. Some of these phases must come one before another, but once the application is up and running and being iterated on, they must be able to work simultaneously. There are five main phases of this process planning/requirements, design, implementation, document/test/deployment, and operate/maintain the system. All these phases were used in the SNHU travel project.

The **Planning/Requirements** phase takes a deep look into what is required for the product to reach its end state. This phase is taking an overview into what is needed and how the plan will be implemented. What planning strategies will be focused on are a huge part of this phase as well as writing user stories, creating the backlog, and planning out the order of the backlog [1]. This is the starting point for the entire process. It is absolutely the most major step and should not be rushed. Estimating how long the entire process will take will not be easy but needs to be broken down as to plan each sprint.

The next step is the **Design** phase. In this phase the team will work out how the program should look in each aspect of the application. This will layout formats for the end users, how the database should look, and how the overall application will come together. The design phase is a next level planning phase. This builds off abstract ideas and puts them in to formattable problems to be solved during the next phase.

**Implementation** is our next phase. This is where the process starts to come alive. All the planning is put into practice. This is where the development team works through the different user stories and other parts of the backlog. The application starts to take shape and the idea goes from being an idea to a product. The product is not ready for production yet though.

**Document/Test/Deploy** this might seem like a lot of steps that should be broken down into distinct phases, but they do not need to be. A well-managed project should be documented all the way through. This means having documentation that explains what is going on in the product when it might not be clear what is happening. Testing needs to be done on every product before it is deployed. The product needs to be put through its paces, and tested thoroughly to make sure it is safe, ready, and stable enough for the end user. If the application is well documented and tested, then it should be deployed. Deployment can be a large step. One might think that once the application is deployed, we might be done, but we are not, we have another step.

**Operate/Maintain the system** is a key phase to this entire process. This means that once the application is live, we are not finished. The agile process means being able to iterate over time. The system will still need to be tested, updated, and regularly checked on to make sure that the goals that were set out at the start are still being met. Once we hit this step the entire process should start over again.

**Waterfall Method**

The waterfall method uses a lot of the same techniques as the scrum-agile method. The waterfall method has requirements, system design, implementation, testing, deployment, and maintenance. The biggest difference between the way waterfall works and the scrum-agile method does is that the waterfall method does not have roles the same way the scrum-agile method does. There is no product owner or scrum master. The waterfall method works much like how a waterfall works in nature. It starts at the top and settles at the bottom. Each step in the waterfall method is held up by the next. There is no working on multiple different areas of the product like in the scrum-agile style. The requirements are laid out, then the system is designed, and next the product is developed. This all sounds straightforward then the product reaches the testing stage. If there is an issue in the testing stage, then the product may have to go back as far as the requirements phase. The waterfall method is not conducive to time management. If all aspects of the product development work perfectly then it is a great method, however that is rarely the case. Waterfall does have its place though and can be highly effective.

**Agile Vs Waterfall Factors**

When developing an application with a team the scrum-agile process may seem like a no brainer. In most cases it is. However, the waterfall method does still have its place. In working on an application solo, one might be more inclined to use the waterfall method as it is next to impossible to do the scrum-agile method alone. If the length of the project is short, and there are not any real possibilities for features to be added then the waterfall method may be what you should go with. With a project that will take some time, have multiple people working together, need updating, and has room for expansion then the scrum-agile method is the way to go. When working on the SNHU Travel project the pace of the project would not have been able to be achieved using the waterfall method. The deadline would have had to of been pushed back several times. The features would not have been as flushed out because the waterfall method does not have as much communication as the scrum-agile style. The scrum-agile style promotes communication at the forefront and has proven itself to be a standard in the industry.

References

* 1. Cobb, Charles G. *The Project Manager’s Guide to Mastering agile: Principles and Practices for an Adaptive Approach.* Hoboken: John Wiley & Sons, Inc.