Final Project Sprint Review and Retrospective

Eric Schooler-Davison

Southern New Hampshire University

CS-250: Software Development Lifecycle

Prof. Cory Thoma

20 October 2024

Applying Roles

For this class I was given a project with an adapting mock company called ChadaTech. The company had recently transitioned to a scrum-agile development team from a waterfall team. This transition was the basis of the experiment as it was used to highlight some key differences between the two. I was run through the various roles and responsibilities of a typical scrum team from the start of the sprint to the end. The three roles we focused on were product owner, scrum master, and dev team.

The product owner is responsible for the software development in that they have ownership of the state of the project. Unlike a manager, the product owner takes a more back seat approach, guiding the team in the right directions but giving each team member the freedom to be their own manager. The position is the deciding factor for the project’s direction and the business sponsor. They do this by being a point of contact for the stakeholder, translating information into the product backlog with the help of the scrum master and the rest of the team. The product owner gives the team direction, pointing them to where they need to be.

The scrum master is the servant leader and ensures that the process is completed sufficiently. They are the scrum experts that follow up to keep scrum understood and enacted. In an ideal world, a scrum master may not be necessary, but people are not ideal. The complexities of day-to-day issues and obstacles fall within their purview as it is their job to eliminate obstacles for the team. They ensure efficiency by keeping everyone on scrum principles. I have personally witnessed what happens when a team is given new tools it’s not used to or does not know how to use. The outcome leads to inefficiency as the tool is half used the new way and half the old.

The development team is committed to self-organization and should not require direct management on how to complete stories from the product backlog. Instead, the dev team completes tasks on the product backlog to create a deliverable. Unlike in older systems, the dev team is one team instead of multiple departments. The team succeeds together or fails together, ensuring that they work together on all issues. Daily meetings give a means of prioritizing and communication to keep all parts of the team on the same page.

# User Stories

User stories take a front seat in Scrum-Agile as they are the basis of creating use cases, test cases, and product backlog items. During initial phases the team intakes requirements in the form of user stories. In Scrum, the stories are used to create complete features that are requirements for the final project. A differentiating factor between Scrum and Waterfall is the completion of entire functions and deliverables after each sprint instead of at the end of the project. User stories are used to create framework for what a complete function looks like and some of the requirements it would have. From this test cases are created, which are criteria to confirm sufficient functionality during tests. During sprints stakeholders may choose to change direction, and with sets of deliverables, the project only requires replacement of the in-question functionality. Alternatively, the waterfall project methodology would only get feedback at the end of the project and would require almost complete recreation of the project.

**Handling Interruptions**

During this mock project, an interruption occurred when the SNHU Travel stakeholders decided to change the project’s direction. Initially they requested a general travel system, accommodating many users. After further evaluation, the company decided to focus on wellness/detox vacations as their projections showed an increase in demand and untaped market potential. In a traditional waterfall approach, the entire project would be derailed and require restarting from the beginning. Fortunately for the team, Scrum was in use which led to a key advantage. Many aspects of the project already suited the proposed changes. The change itself was the type of vacation to focus on, but as vacations were still the core principle, much of the project was perfectly suited to this change. This is a big part of why Scrum-Agile is preferred, as markets change regularly, and companies must adapt frequently. Making adaptation a core principle of Scrum where interruptions are expected and not avoided saves a lot of time and productivity.

**Communication**

A key communication point for this project was an email as a developer to the project owner for how to adapt test cases for updated user stories after the aforementioned project directional shift.

*To: Christy*

*Subject: User Story Clarifications Dear Christy,*

*I have looked at your user stories and am developing test cases for the different features to determine whether the product passes or fails. I need a bit more detail so that I can use more specific metrics to clearly define my test cases. Can you answer the following questions for me?*

*User Story One*

*• Where will the list generate from?*

*• Should the list count down from 5 to 1 or start with #1?*

*• Will the list integrate a top list from elsewhere and pull from that class to generate results*

*User Story Two*

*• What kinds of column headers would you like to see for this option?*

*• How would you like the deals to be organized? From most recommended or from most people who like XYZ like this?*

*• How will price range affect these two lists or will price range be its own list?*

*• Do the sales need to have a description like the other entries or should it be a link to the main entry of the destination?*

*User Story Three*

*• Will price limit be introduced by selecting premade ranges such as 100 – 500, 500 – 1000, etc. or will it support exact number entry from the user?*

*• How will the filter apply to other parts of the system?*

A good leading principle when communicating is doing so effectively so the most amount of information can be gathered on the first pass. To get the right answers, one must first ask the right questions. When communicating, a good method is to stick with who, what, where, when, why, and how questions as much as possible as these prompt the most response and don’t result in yes or nos. My email encourages further communication, asking clarifying questions alongside broader digging questions that create discussion amongst the team. While it is impossible to get all the information at once, knowing what you need to know now makes all the difference when asking for directions. It’s not possible to overstate the importance of good communication, as it is an easy way to save time and keep the line open with peers.

**Organizational Tools**

In addition to communication lies the tools that assist that open line. Such tools can be broken down into collaborative tools such as Jira, and events such as scrum meetings that happen throughout the project. The two tools are powerful apart but shine when used together. It gives everyone a space that lets all members see progress, product backlogs, meetings, timeframes, sprint progress, etc. When combined you effectively have a one-stop shop for all the information everyone needs. That creates a completely open line of communication where the entire team can see what everyone is doing and assist each other while getting reminders for Scrum events like the 15-minute stand up or backlog meetings.

**Evaluating Agile Process**

Scrum-Agile is by no means perfect, while it offers much needed improvement from the Waterfall method, different projects may benefit more from sticking to Waterfall. Agile is flexible, giving it the adaptability needed to change as the project goes on. Their focus on delivering value quickly is one of the big positives, but it is not without its challenges. One issue is the potential of scope creep, where requirements adapt too much and the team loses focus, adjusting frequently but without a vison that makes sufficient progress. This is in part since much of the planning is not done upfront, but as needed. Scrum is also not as suited to large teams, and benefits most from small groups. Communication becomes a bigger and bigger problem the more people you must communicate with. Scrum teams may also become fatigued from the number of meetings as it can easily become excessive and take time out of the day while not adding any value to the project. Because of these difficulties, Waterfall may be a better approach if your project is very large, complex, or non-linear.

In the case of SNHU Travel, the team is small, requiring only one scrum master and product owner. The project itself is not overly complex and progress can be measured easily by means of deliverables, this means it is not overly complicated and costs little to change direction. My assessment is that Scrum-Agile suits SNHU Travel well as the project lacks the size and scope that makes agile less than effective.