Throughout the conduct of this sprint, I was able to experience the Scrum framework from a variety of unique perspectives. At the onset, I was tasked with creating a team charter, taking on the role of Scrum Master. As the Scrum Master we were tasked with completing a team charter, to set the direction of the group and state the needs of the work being performed. The Scrum Master is uniquely responsible for upholding the framework itself and answering any questions the team may have regarding various Scrum deliverables and processes. The Scrum Master also leads the Daily Scrum, in an effort to facilitate communication between the other roles on the project. Finally, the Scrum Master will lead the team in the Sprint Review and Retrospective in order to apply lessons learned and communicates these throughout the organization.

After the team charter was completed, our role shifted to that of the Product Owner, as we were tasked with collecting and analyzing the dialogues that we have had with our users, and developing User Stories to share with the team. These User Stories become the basis for work that is to be performed on the existing code. This is one way we worked to accomplish our responsibility of the Product Backlog as Product Owner. It would seem that perhaps the most important part of the Product Owner’s responsibilities within the team is to facilitate healthy dialogue with the Users, and organize the various wants and needs into a respectable to-do list, or product backlog. The overall direction of the team is ultimately decided by the voice of the customer.

Next, I took on the role of Developer. The Developer role is unique within the framework, as the various efforts of the team are in direct support of the Developer, but the Developer does not lead the project itself. I find this to be a very different approach to accomplishing work, and in this specific circumstance, it best serves those closest to the work being performed. As the Developer, we worked to add specified functionality to a list that was being sent to our consumers. This work was technical in nature and required us to work with some prebuilt Java programs that were then output to an executable JAR file. As the Developer, we were responsible for reviewing both the User Story and current codebase in order to problem solve a solution, provide estimate for implementing that solution, and then execute that solution within the bounds of the sprint.

The true benefit that was seen throughout this specific application of the Scrum-agile approach was a clear direction, from the development of strong user stories. The user story provides direction and achievable outcomes for the team, comes directly from the consumer, and undergoes the refinement of applying the framework to the content of the stories. This level of detail and refinement comes from having pre-defined workflow, nomenclature, and deliverables, which all come from the Scrum framework. Another benefit to the Scrum-agile approach is the consistent communication by the team, in order to problem-solve anything that might come up during the performance of the sprint. Had there been any challenges in accomplishing the functionality, the team member would have a great opportunity to discuss this with the team each day.

The ability of the Scrum-agile approach to work through emerging changes to requirements is perhaps the greatest strength of this methodology. We see this clearly during the course of this sprint, when requirements for the executable JAR file changed suddenly. The changing requirements were quickly addressed with the team, and resources were able to be shifted to meet this emerging need. Within my own experiences, this is unique to the Scrum-agile approach simply by default. The predefined team structure and daily scrum play a huge part in this inherent flexibility.

The following is an example communication that took place between myself as Developer, and the Product owner:

“Greetings Product Owner,

It is my opinion that the development team could benefit from a wider range of input from our user base. This could take the form of polling or email marketing. Currently, we have a large amount of homogenous stories that deal with a specific type of functionality. In order to expand on the site’s capabilities, we need to identify more user stories.

Thanks”

This communication is a great example of consistency in communication, because the Developer knew exactly who to contact with this issue, and what action they might take. Again we see the benefit of a set of shared nomenclature and specificity in purpose.

I would say that perhaps the most effective of Scrum tools that were utilized by the team during the performance of this sprint are the well-defined user stories. Without these user stories, the developer would have a lack of direction and the team would not be able to accomplish a large piece of functionality in a short window. This specificity of purpose is the idea behind what makes up a single sprint. Without a well-defined goal in mind, the team flounders and wastes time and energy in deciding what to pursue.

In conclusion, the Scrum-agile approach was extremely effective in this sprint. The Team was able to add significant functionality that was specifically a request of the consumer. The key to the success of the team was a well-defined user story, and a structured approach to communicating requirements, in support of the work being accomplished by the Developer. A possible con of this approach would be that the end user might become overly fond of the quick response time provided by the Agile team.