SWEN 732 - Group 4
Final Report
Ian Adler, Tyler Borrelli, Swetha Sajjala

Introduction

Group 4 sought to build a Point-of-Sale application to allow restaurants and other stores to make sales from a catalog of configurable items. This project was intended to be small enough that it could be completed in less than a semester and leverage a GUI, but no web connectivity per external stakeholder restrictions. The main focus for the group this semester was to focus on the Essence Health and Status tracking system from the Object Management Group to get familiarity with it through hands-on application. The group's secondary focus was process engineering to derive a process sufficient for development of the application. The tertiary focus for the group was to successfully develop an Minimum-Viable-Product for the application that could support sales of custom items at a minimum.

Alpha Status and Highlights

Alpha state tracking is an essential tool for keeping track of the various stages of a software project, particularly during the development phase, to ensure that the project is advancing as planned, and any potential issues are identified and addressed promptly. Alpha state tracking can be a crucial element in the development of many software applications such as restaurant application in this case to guarantee the project's success and satisfy user needs.

Monitoring the project's development entails keeping track of any challenges that arise and the solutions put out to address them. The development team can quickly detect and deal with any difficulties that arise because of alpha state tracking, which guarantees the effectiveness and efficiency of the development process. This in turn prevents project delays and guarantees that it is completed within the anticipated spending limit and timetable.

Monitoring the project's alpha state helps the development team make sure the program satisfies the requirements for reliability and functionality. Additionally, it enables the team to spot potential problems and take swift action to fix them, guaranteeing that the application complies with all requirements. Tracking alpha states is also important for assuring the final product's quality.

Alpha state tracking aids in the delivery of a high-quality, reliable, and functioning solution for restaurant applications, satisfying consumers' needs and adding value to the company.

Alpha State Tracking :

Our team used to have frequent meetings on the alpha tracking part from the beginning. We used to track each alpha and its state in every meeting and we used to have agendas for each meeting about what should be done by the next meeting. We would post the minutes of the meeting, when the next meeting should be and the agenda for the next meeting. This made our work simpler as it tracks every task and helped during the alpha state tracking.

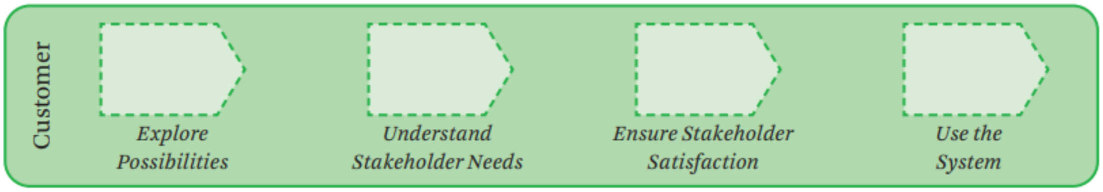
Activity Spaces:

Software development using Essence focuses on a variety of issues, including customer solutions, opportunities for growth, and team support. There are activity spaces, which resemble groups of linked tasks, within each area. The work that needs to be done to develop software is organized by these activity spaces, which also make it simpler for users to find and employ self-help techniques.One activity space, for instance, "Understand the Requirements" , focuses on finding out what the program must accomplish and what its users want it to do is the main focus of this activity area. This involves a variety of tasks, like speaking with people and developing prototypes. By completing these tasks, the group can progress towards achieving the desired alpha states, which are like milestones that show how well the team is doing.

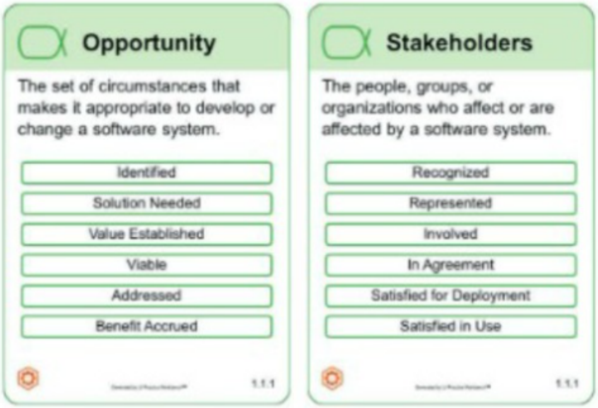
"Implement the System" is a different activity space. This is where the team actually creates the software's code. This involves a variety of processes, such as developing the architecture and testing the code. By carrying out these tasks, the team can move closer to the code's desirable alpha states, such as ensuring that it functions well and is simple to maintain. Essence's activity spaces assist teams in producing software in a consistent and structured manner. Teams can work to achieve the appropriate alpha states and ultimately produce effective software systems by adhering to the activities in each activity space.

Status and progression

When we consider the Customer perspective, we have a set of activity spaces, such as exploring possibilities, understanding stakeholder needs, ensuring stakeholder satisfaction and using the system.



To address all these activity spaces, we have two alpha states : Opportunities, Stakeholders
We have a set of alpha state cards to progress with all the activity spaces in each alpha



Opportunity

The set of circumstances that makes it appropriate to develop or change a software system.

- Identified
- Solution Needed
- Value Established
- Viable
- Addressed
- Benefit Accrued

Stakeholders

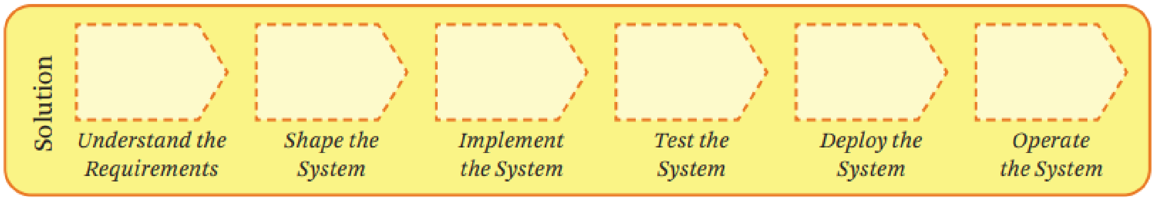
The people, groups, or organizations who affect or are affected by a software system.

- Recognized
- Represented
- Involved
- In Agreement
- Satisfied for Deployment
- Satisfied in Use

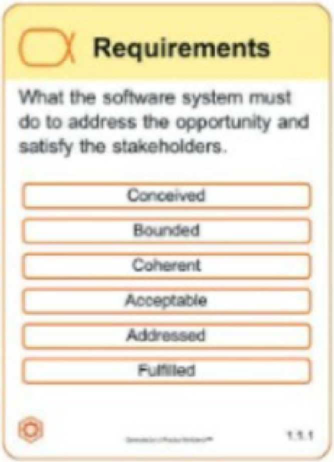
During the progress of the project, we started with the opportunities identified in the beginning. Later we discussed the problem statement and what should be the solution needed and if there is any value established through the opportunity we get by doing the work. Also, we have concentrated on if the opportunity that we get to work on this is viable or not.

When it comes to stakeholders alpha, we had a little confusion on this for 2-3 weeks at the start. We were unable to know what exactly this stakeholders concept represents in this team project. As it is only a group of three, we had a doubt that how will we three only be the stakeholders and how will it impact the software development. Later, we have recognized the stakeholders are not only the people who will develop the software system but also people who will use the software and invest in the system. Based on those stakeholder’s feedback, which involved them in the process and made their points in decision, we worked for the development of the software system.

Moving onto the Solution perspective, we have the activity spaces mentioning Understanding the requirements, Shape the system, Implement the system, Test the system, Deploy the system, Operate the system.



Solution perspective concentrates more on the requirements to develop a software system and development of the software system. In this, we had first understood the requirements based on the opportunities we had and stakeholders requirements. Later we had designed a basic architecture of how the system would be and how we should design it. Based on the other alpha “ software system” progress, we have implemented the system and tested it before deployment.
The two alphas in this category is : Requirements , Software system



Requirements

What the software system must do to address the opportunity and satisfy the stakeholders.

Conceived

Bounded

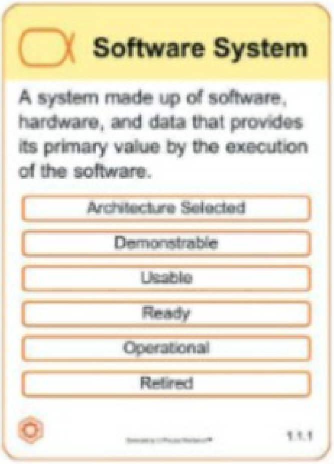
Coherent

Acceptable

Addressed

Fulfilled

1.1.1



Software System

A system made up of software, hardware, and data that provides its primary value by the execution of the software.

Architecture Selected

Demonstrable

Usable

Ready

Operational

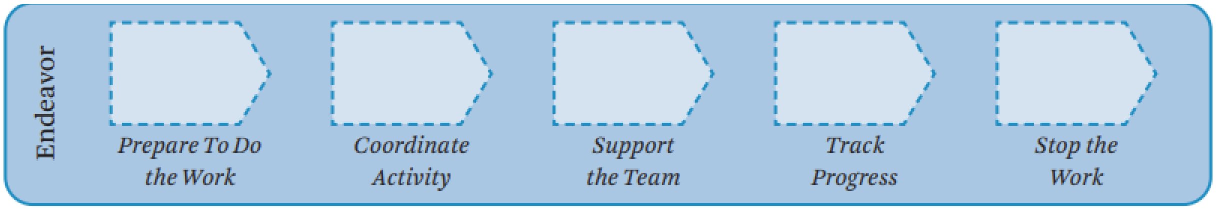
Retired

1.1.1

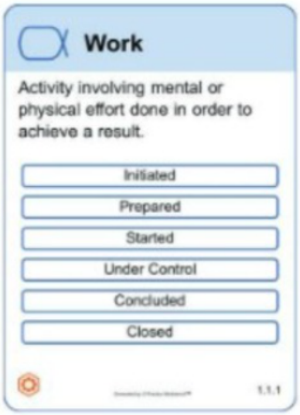
First, we have identified all the requirements and ensured that the requirements are conceived, bound to the software system we architectured. Also, we had to check if all these requirements are coherent and acceptable to develop the desired software system.

When it comes to software systems, firstly we had finalized architecture based on the brainstorming ideas. Then we made sure that the system is demonstrable by working on the functionalities. Based on the endeavor alphas, we made our software system usable, ready and Operational.

Moving onto the Endeavor perspective, we have the activity spaces mentioning Prepare to do work, Coordinate activity, support the team, Track progress, stop the work.



During the process of the project, we thought the endeavor alphas have a crucial role to progress. The alphas team, work and way of working helped us to be successful in regards to this project.



Work

Activity involving mental or physical effort done in order to achieve a result.

Initiated

Prepared

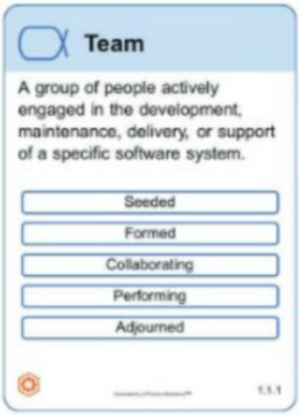
Started

Under Control

Concluded

Closed

1.1.1



Team

A group of people actively engaged in the development, maintenance, delivery, or support of a specific software system.

Seeded

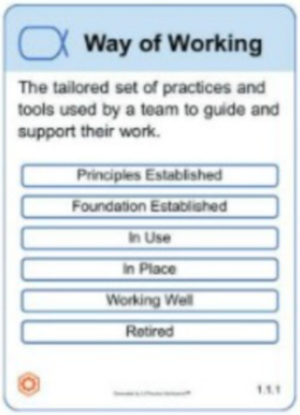
Formed

Collaborating

Performing

Adjourned

1.1.1



Way of Working

The tailored set of practices and tools used by a team to guide and support their work.

Principles Established

Foundation Established

In Use

In Place

Working Well

Retired

1.1.1

First, we focused on the team formation and got to know each other by the personality test. Later the frequency of meetings we had and the involvement of everyone in the meetings made us get to know each other better. Later, the team work along with its way of working made us build the software system effectively.

In the beginning of the project, we had concentrated on the stakeholders and the opportunities based on the requirements needed. Later on, our focus shifted onto the software system. It took us more time to progress from one alpha state card to the next in case of a software system. To work on this software system, we had to split the whole concept into sub tasks and used the SCRUM process to progress. We used Trello for tracking the sprint plan and how the work is going.

Finally, by the end of the semester, we were able to use the essence practices in the development of a software system involving the alpha cards. The current alphas we are left with are the opportunities in theBenefit accrued state and requirements in fulfilled state.

Process design and use

The project scope for this semester was small given the primary goal of focusing on using Essence to track the Heath and Status for the Project which made focusing on a small-scale project more beneficial. Additionally, the project scope was influenced to be smaller due to schedule pressures given the length of the project, less than one semester, the amount of time each member could contribute given outside commitments, and personnel concerns given the small size of the team, 3 developers. The scope of the project resulted in a couple requirements for the process that was to be designed for the project. First, familiarity with the tools and techniques selected as part of the process was important given the short time for the project and the amount of time members would have available, there would not be a lot of time available to learn brand new tools and how they tie into the process. Second, ease of implementation was important for the process due to the time constraints, so all tools should work "out of the box" and should not need a lot of configuration and time to set up to avoid ruining the schedule. Lastly, the cost necessary for any tools selected was a big requirement for the process since the project was only scoped for less than a semester so any tools should be free for at least that timeframe, but ideally free forever.

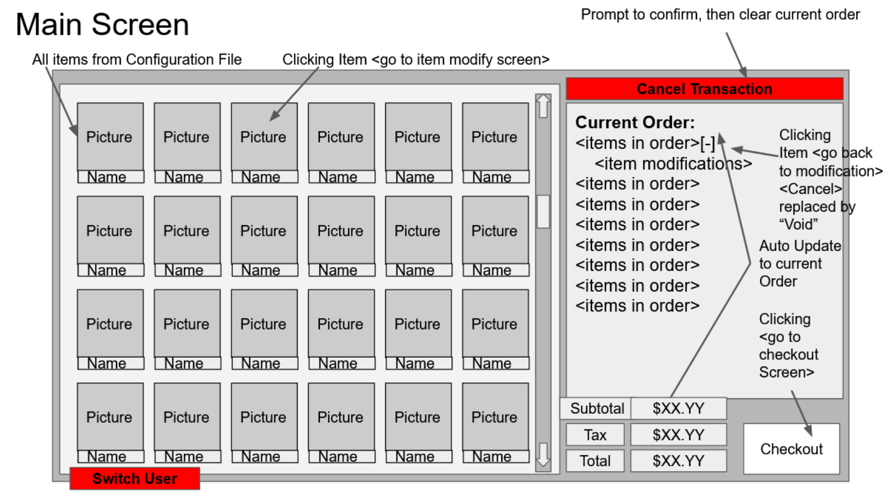
The process for the project was grouped into 4 logical categories: Change Management / Version Control, Development, Collaboration, and Management processes. For Change Management and Version Control the process was to use git version control for all version control needs and to use GitHub as the remote source for the repository. Git, remote repository hosted on GitHub. These version control processes satisfied the requirements for the process since git was the version control solution that 2 of the 3 developers were most familiar with and the third developer had no prior version control familiarity, so this was the most familiar solution. Additionally, both version control choices were fully setup and configured out of the box and are free to use. For Development, the Visual Studio Code IDE was selected as the integrated development environment for the project. This was selected because all three developers had used it previously so there was familiarity, it was fully configured for Java support with the installation of one simple plugin, and it was free to use. The Java Program language was chosen with Maven as a build management solution due to familiarity for the developers. JavaFX was selected as GUI Framework, no developer on the team had familiarity with a GUI Framework so there was no better choice to meet the requirements for the process. Development and committing to version control was done by working in feature branches for all new development and then requiring a code review approval needed before merging. Collaboration for work to be done for the project was supported by using Discord for voice and text communication (using separate channels for specific topics, ex. Meeting planning), using Google Drive shared folder for assignment and misc. document collaboration, and using Trello for tracking Product / Sprint Backlogs. These collaboration platforms were all familiar to all developers for the project, ready to use immediately with little configuration and free to use so they fulfilled all necessary requirements. Lastly, Management processes consisted of Essence for Project Health and Status tracking and Scrum for Project Management. While no developer had Essence familiarity prior to the project, it was a requirement from the external stakeholders to track health and status using Essence and the goal of the project was to gain exposure to this, so it is overlooked for the familiarity requirement for the processes. Scrum is the project management framework that the team had the most familiarity with, so it met that requirement.

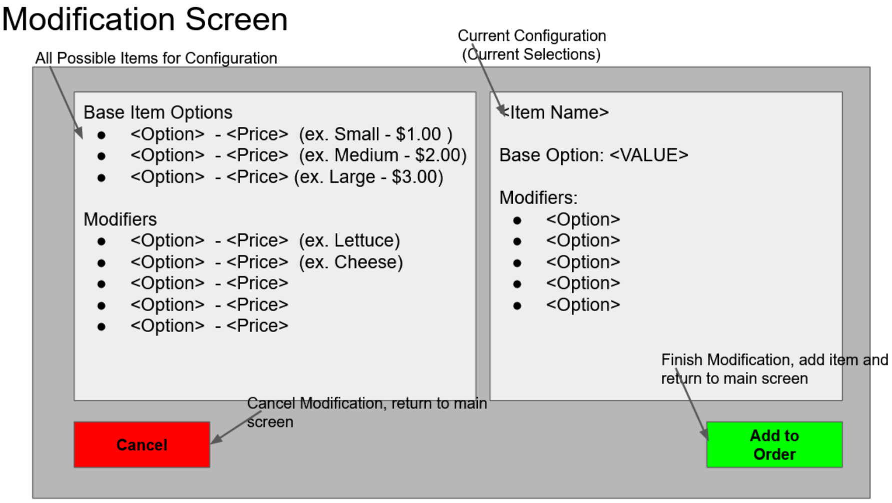
One of the process items that worked the best for the team was Discord for communication. It allowed the team to work asynchronously since schedules did not always line up so the team could communicate, and it allowed the team to have meetings virtually when in person would not have been convenient (ex. late at night). Specifically late-night collaborative programming sessions where members were available for 2 hours at a time all online at the same time to work on their respective tasks and discuss any issues immediately as they arise as well as share knowledge. One aspect of the process that did not work well was the code reviews for collaborative development that were required prior to merging code to the main branch. The tasks were divided into very small efforts so the code that was submitted for review was often not complicated so there was very rarely feedback provided and the reviews felt more like a formality than a legitimate inspection of the quality. The main takeaway from the process was specific to Scrum Ceremonies and that was that while the ceremonies need to be done (ex. Sprint Planning, Sprint Review, Sprint Retrospective, and Daily Standup), they can be done in a way that accommodates the team and the project. For example, the Sprint Review and Sprint Retrospective can be a combined meeting talking about the respective topics back-to-back, and for smaller sprints where all developers aren't necessarily full-time, they can be abbreviated, but still beneficial.

Product design and development

The product that was to be designed was a Point-of-Sale application that can be used to make sales, with the targeted use case of restaurants. An important requirement for the project includes that it was offline and locally available only due to requirements imposed by the stakeholders. Additionally, due to schedule concerns with the length of the project, the requirement was imposed for single Redundancy only, namely that there are no backups or failovers in the event of a failure, if application crashes it fails in order to be simpler. Additionally, there is a requirement that the items that are available for sale be provided in a configuration file so they can be changed without requiring recompilation, but not requiring a database to allow for a simpler design. There was a requirement that the user is able to remove or modify a single item without stopping the whole order as a quality-of-life improvement, but also a separate requirement that the user have the ability to remove the whole order at once (ex. a customer leaving). While the items can be changed with the configuration file, there was a requirement that the layout of the application (relative arrangement of UI items) be fixed. Lastly, there was a requirement that no credit card processing would be tied-in to the application and all processing would be "mocked" by asking for the card number and details then not processing anything to avoid needing to comply with credit card regulations.

The Product architecture centered around the JavaFX GUI framework and focused on a main screen as the entry point to the application with widgets and secondary screens to support all required functionality. Configuration for what items are available for sale was handled using a configuration file in the JSON file format specifying all items available for sale and details for each item. Screens were used to configure the window to a more targeted layout specific to the screen needs. Screens included in the project include: The Main screen to show all items and items for the current order, the Modification screen to configure new items before adding to the order or to modify items already in the order, the Checkout screen to confirm details for the current order prior to the customer selecting a method of payment, and then Cash/Card checkout screens which supported the customer paying via cash/check or credit/debit respectively. The design documentation for who the layout of the screens should be implemented was captured using a wireframe mockup of the application due to the small scope of the project and the team's goal of focusing more on the application of Essence to track Project health, rather than focusing on development and documentation. For examples of the wireframes for the Main screen and Modification screens see figures below.




Modification Screen
All Possible Items for Configuration
Current Configuration (Current Selections)
Base Item Options
<Option> - <Price> (ex. Small - $1.00)
<Option> - <Price> (ex. Medium - $2.00)
<Option> - <Price> (ex. Large - $3.00)
Modifiers
<Option> - <Price> (ex. Lettuce)
<Option> - <Price> (ex. Cheese)
<Option> - <Price>
<Option> - <Price>
<Option> - <Price>
<Item Name>
Base Option: <VALUE>
Modifiers:
<Option>
<Option>
<Option>
<Option>
<Option>
Finish Modification, add item and return to main screen
Cancel Modification, return to main screen
Cancel
Add to Order

Product quality assurance for the project was assured by having well defined acceptance criteria for every activity so that what's expected for the final solution is understood and by forcing code reviews of all code before merging.

The design activity that worked the best for the team was the wireframing exercise. This allowed the team to visualize the application early with minimal effort to see what layouts and behaviors would work or not. It also established the layout of the application early so that later tasking could reference the wireframes and be consistent. The product development aspect that the team had the most issue with was laying out tasking such that team members could continue to make progress while one member was finishing their current task. There were many instances when tasks for a sprint were connected in order to tie into a common sprint goal, which resulted in one task holding up another task. The primary development lesson learned for the team was the benefit of having spikes for new technologies. No developers on the team had familiarity with the JavaFX framework used for the project, but there was not a spike set aside to explore this technology beyond simply setting up the project for a basic application following a tutorial.

Instead, the developer who took the first task that required GUI design beyond the simplest application had to figure out the framework and then teach the rest of the developers.

Team collaboration

Collaboration is an essential part of the proper functioning of a team. Our team in particular had several requirements, the biggest one being that we needed to work asynchronously. This includes both working independently, as well as virtually. We are all busy individuals with other tasks on our plates. Another important collaboration requirement was that when we did meet, it needed to be in the late evening. Between classes and full time work that takes up most of the day.

We very early on decided on borrowing some practices from SCRUM. Multiple of them were for collaboration. One that we used very often was the daily standups, but with a twist. Traditionally Standups are done literally standing up. Since we all operated on different schedules we opted to do it through a text channel. At the beginning of the semester we defined a template to be used for everyones daily status. It included what the individual did the day before, their plan for the day, and if anything was holding them up. We also tracked tasks using sprint that were two weeks in length. The sprint board was laid out like a typical sprint board. We had a project backlog containing all of the tasks to be done over the course of the project. Then a sprint backlog, which was the specific tasks we wanted to accomplish over the course of the next two week sprint. Then the last two columns were for in progress tasks, and then finished tasks for that sprint. In the beginning we went through and assigned story points to each task. Then while doing sprint planning we picked an appropriate amount that we thought we could accomplish over the course of two weeks. Another SCRUM practice we used that played into the Sprints was Reviews and Retrospectives, again with a twist. These are traditionally separate meetings as they have different tasks. Either to review the work done over the last sprint, or the process over the last sprint respectively. However, since time was a premium we squeezed these together into a single meeting. Which was successful as we kept the two topics separate. If there was anything that was not accomplished in the sprint, it was rolled over into the next.

Several different technologies were used to facilitate collaboration within our group. Being that we all lead different lives through this course, we were forced into remote style work. For communication we used Discord. Within our Discord server, there were several different text channels each for a different purpose. They are listed below

- General discussion
- images/links to Essence related information
- Meeting planning
- Google drive document links
- JavaFX resources
- Daily stand-up statuses
- Story Pointing

Another technology used was Google Drive. This was for any Work Products created by our team either as part of the project or general homework assignments. Lastly, we used a Trello board as our sprint board.

There is little to be done to improve this collaboration process other than losing more sleep. For next time, given the size of the group and the length of time we had to accomplish our goals, SCRUM and sprint based development probably is not the right move. A more traditional waterfall approach would have suited us better. We only picked our SCRUM like approach because of the heavy push from the class material and a desire to mimic what exists in the world.

Even with our abridged version of SCRUM, we found it difficult to stick to the schedule and ceremony of the collaboration process we did have. The one that was most apparent was the daily stand-ups. Since we all were constantly busy, it was not hard for the messages to be missed here or there. A lot of them just stating that nothing was accomplished certainly made the importance of them feel less. To a lesser extent we also had sprints of varying sizes. The average was certainly two weeks but we only really hit it once exactly. However we were very good about doing exactly what we said we would do. After our initial brainstorming session, we quickly realized that we had way too many ideas for the short amount of time we had to accomplish them. So, we pared down a manageable amount. Along with that, we did a good job of maintaining the same velocity over sprints. These kind of are the same idea, we know how much work we can do and did exactly that. A collaborative technique that was very positive for us was pair programming in the beginning. When we had little idea how to use the architectures that we had selected, we worked together to figure it out. As we became more comfortable with Java and JavaFX these pair programming sessions turned into collaborative programming sessions. In these sessions we essentially each worked on our own tasks but at the same time in a voice call. This allowed us to easily ask questions as they come up instead of having to wait for a meeting or the next daily standup.

As previously mentioned, sticking to the rigorous ceremonies of SCRUM is extremely difficult when working on this project is the tertiary priority for all of us. In the future just having the project be a higher priority would help this. Also, most of our pull requests were symbolic. We largely knew what the others were doing and took their word for it. On a project this small that usually goes fine. It would have been better practice to do a more through and harsh code review on each pull request.

<u>Conclusion</u>

Group 4 was able to successfully meet all the goals set out at the start of the project: learning Essence, deriving a process, and completing an MVP. The team had a running Alpha Status that was reviewed weekly (or minimum every other week) to track the current health for the project. This tracking included discussions on why the group thought it was in a particular state supported by what activities had been completed and work products were generated which gave all members of the group good exposure to Essence. The team was able to generate a process and follow through on that process without incident and the group was able to successfully complete and demonstrate the app MVP minimal feature set. With all goals met, the project is considered a success and a good learning opportunity for all members of the group.