# Group Project
# Shortest (cheapest) Path - Project Documentation
# Data Structures and Algorithms (CS20230)

Thomas Anderson
17202065

April 30, 2018

I have broken down my journal into three sections. These three sections were the original three sprints that we had anticipated. In general, I feel that I have learned a lot about software engineering while doing this project and Ifeel more aware about potential issues for future projects.

This was my first software engineering project where a framework like scrum has been used. It is also important to note that the members of our team are all new to software engineering and software developmeent in general. Being new to software engineering means not having seen the issues that come with project planning, sprints, time management for development and project communication.

## Sprint One

We started the week off with lots of enthusiasm. We spent the first day trying to get setup with the various project tools that we would be using. This was chalellenging because we weren't familiar with the majority of the tools that we would be using.

We spent a lot of time trying to setup our sprints. We probably should have done this before starting the first sprint because it it took us quite a while to do this. We weren't sure how to use Trello and we had heard that it supported burndown chart functionality, but it's not clear how to use it. We eventually discovered that we could use an add-on called Scrum by Vince to obtain scrum burndown chart functionality, but it was not an easy task to find this. I really think that Trello is a tool that could be improved.

The key to being successful with burndown charts and actually getting value from them is to put effort into keeping the product backlog and spring cards (in Trello) regularly updated and accurate. I found it quite difficult to really estimate how long a task would take me. Some tasks took less time than Ihad expected and others took significantly longer. I will be careful about keeping the cards and times regularly updated for future projects.

During our first sprint we:

- Created schedulers for scraping data from OpenWeatherMap and JCdecaux
- Created class methods to parse the scraped information
- Created an AWSRDS Instance with tables for the OpenWeatherMap data and JCdecaux data
- Created a product backlog and setup Trello for burndown charts

We implemented all of these features, but what we realized afer the sprint had finished was that we had to continually tweak all of these items that we thought were finished. For example, we initially weren't completly sure which weather data we needed from OpenWeatherMap so when we changed what we wanted, we had to go back and modfiy the scrap, parse and SQLmethods to get what we needed. This took considerable time. I think we would have been better off had we put more effort into longterm project planning and figured out exactly what we would need later in the project.

The most challenging technical aspect for me this sprint was to learn about object-oriented programming. Iam very happy that Iput in the time to create classes because it allowed me to easily share variables between classes and separate my methods(functions) by class name. The object-oriented features that I worked on (schedulers to request, parse and exectute SQL statements) were challenging because it was the first assignment for me this year where I used object-oriented programming. However, I'm really glad I put in the effort to learn object-oriented programming for this project because it is already proving

to be useful for other class projects this semester.

## Sprint Two

Sprint two was a lot more challenging than sprint one because during sprint one we had the luxury of being on study break which meant that we could focus our time on completing our sprint tasks. This time we had much less time to spare and much less time to learn. This week we weren't only doing software engineering. We also had assignments for other classes and we had to be in class all day. This meant that we only had a few hours in the evenings to work on our project.

Some of the tasks that we worked on during sprint two are below:

- Created functions for SQLinput statements to send information to the database
- Create a basic flask application
- Setting up ec2 instances for a webserver and an application server
- Create methods to retrieve data
- Setup Nginx
- Application Server error handling
- Flask - Google Maps

We had a problem with the way the api schedulers(from sprint one) were working. We had written them without considering error handling. This meant that when one of the methods we were calling using the scheduler encountered an error it would crash and our data stopped scraping. This was a big problem and it took us quite a while to understand the cause of this error and implement a solution.

It took us a long time to understand how Flask works and more specifically how to connect Flask to our database and Javascript. Since it was the first time we were relally using these technologies together, I think we would have been better off if we planned to allow for more time to accompish these tasks.

We also had an issue where our Amazon WebServices account ran out of credit so our RDS instance was deleted. This caused us to lose a lot of time because we had to figure out why this happened, create another database and update our application to use the new database.

I think given the time constraints, we were able to get some very useful things done, but we didn't finish all of the tasks we had planned too which meant we had more to do the following week. I think its' very important when planning for each sprint to choose tasks that are realistically achievable with respect to time constraints.

## Sprint Three

Sprint three was more difficult than we had anticipated it would be. We had not finished all of the features from the second sprint so we had to continue to work on those features and put off working on the spring three features. Thanksfully though we had anticipated that we may need extra time so we had time to extend sprint three by another week. Unfortunately this wasn't enough for us to implement all of the features we wanted to.

Several factors played into us not finishing our goals for the second week. We had an issue with our AWS RDS instance where the accuont attached to that instance ran out of money. This happened because the account wasn't properly setup as a student account and it had multiple instances running at the same time. We probably should have foreseen this kind of problem and planned accordingly.

We also had various issues with requesting data from API's and inserting it into the MySQL database. I had not foreseen the need to implement exception handling for the scheduler that I had created. It took a while to debug and fix an issue that we had with this.

We also had an issue with the way the scheduler/scraper program was run because we decided to implement the scheduler/scraper as an installable Python package. This worked well once it was setup, but Iwish we were given a bit better guidance and perhaps spent more time thinking about how to implement this beause it would have been easier to user something like a chron job or more interesting to create an actual daemon. The solution we impelemented used a bash command called screen. This works well, but it wasn't ideal.

We ended up implementing most of the features by the deadline, but with more planing and experience I think it could have been a smoother journey.

Another issue that we had during sprint three was that we had not developed and deployed to our webserver so when we deployed the flask app to our EC2 webserver, we saw that the database connection wasn't working. We used SQL Alchemy and this was a bad decision because we have since read that it isn't a good tool to access the database with. We would have been better off had we used MySQLConnector. Unfortuntely, we weren't able to refactor our Flask app in time to have the database connection working on our EC2 instance.

## Personal review / Retrospective

In general, I think this project was time well spent. I feel like I have a better grasp of the challenges that come with working in a software engineering team. I only wish Ihad more time to complete more of the features.

It was much more difficult than I had anticipated to coordinate between us to accomplish the assigned work. It became quite apparent that we each had very different strengths and weaknesses and that we would have to collaborate together in order to be able to finish the assignment.

The standup meetings were new to me and in general I found this to be a very good way to understand the progress and challenges the project was facing. Ithought it was quite useful that each sprint had a different scrum master. This gave us all a chance to experience being the scrum master and also not being the scrum master. While I found all of the project organization to be very useful, I found that it was actually quite difficult to setup the project organization. Trello was a useful tool, that made managing the cards easier than it would otherwise be, but Trello was not a very good tool for burndown charts and the product backlog. We ended up having to find and install a Trello plug-in called Scrum by Vince. This was useful, but definately not ideal. I would guess that the paid version of Trello is better, but there really should be a free tool for managing the product backlog, cards and burndown that has all of that functionality as a standard feature.

The sprints were also very useful for breaking the project up into small achievable tasks. This is very important when facing challenges where you may not have a full understanding of the task at hand. It also allows you to communicate what you are doing with your team memebers in an understandable and efficient manner. I really like the sprint approach to project management. I think it will be interesting to see how it is done when working for several different companies.

I found the project to be quite challenging and a little overwhealming because of the number of new tools we were exposed to during the sprints. For this project we used Trello, Git, Amazon AWS, Slack, Google Charts, OpenWeatherMap JCdecaux, among others. I thought it was very valuable to have gotten some real exposure to these tools. I particularly liked using AWS, slack and the different API's.

It was also very difficult to complete all of the features for this project due to the time constraints. We have four other classes this semester and all of them have various assignments due throughout the life of the project. Time management will probably always be an issue for software projects so this was probably quite good preparation for the real world. I have worked in industry for a while and it's normal that there never really is enough time for everything.

However, I wish I had more time to practice working in Flask and JavaScript. While I had some exposure to Flask, I was mostly focused on the working with the database, writing SQL, requesting and parsing the data.I would like to work more withFlask in the future because I think it's important to be comfortable with creating front ends for applications. It's important to be able to have something to show for your work. JavaScript and Flask could be very useful skills to have for showing recruiters or hiring managers in companies, who may not have technical computer-science knowledge, what a project does. I will try to learn more JavaScript over the course of the next year.

I am really enjoying software engineering because I think that these projects have given me an opportunity to learn a lot in a very short period of time. I was able to get to learn a lot about various aspects of technology, but probably more importly project management. Having finished the software engineering group project, Ican see why project management is so important for a project's success.