**Team Umbrella - MileStone 1   
Sean Skinner, John Adams, Stellan Giffin, Andrew Fritz, Jose Sainz, Edna Mutonga**

**Key Tools**

* Git Hub
  + GitHub is a distributed version control software that will allow our team to share our work from our local machines onto an online repository where each team member can make changes to the project. With this root repository we can each download a version of our project and work on the same or different features simultaneously. It is incredibly efficient since it only tracks changes made to the software as opposed to saving multiple instances of the same project. Since they provide free internet hosting, we won’t need a server to store all of our progress.
* Google Docs
* Google docs will help with real time collaboration that we could use to edit and create documents and presentations. Google docs was our choice because it enables each team member to input their collaborations simultaneously and make changes instantly. Google docs is also accessible on a website, so it would make it simple for any of the team members to access the google internet servers from anywhere. This feature allows us to create all of the same features provided by Microsoft and can be interchanged if needed.
* Discord
  + Discord is a service we plan to use to effectively communicate as a team remotely. It has multiple features for us to use such as voice rooms where we can talk to one another, message boards where we can easily communicate with each other when we cannot all be on at one time. Discord also allows us to share files and stream our screens to one another. All in all, Discord is a must have tool for communication in the digital age that will keep our team connected and informed.
* Visual Studio 2019
  + Visual Studio is a software suit developed by Microsoft to aid in the quality of code produced in its environment. With its multifaceted support structure as well as broad industry adoption such tools are critical parts of a design process. With built in Git functionality the capability for team collaboration and version control is streamlined.
* C/C++
  + C/C++ are low level languages that allow the programmer to be more hands on with the data. Most of the data we will be handling will be in CSV format, allowing for relatively easy processing regardless of the language. The biggest challenge here is managing time complexity and storage. The CSV files will be upwards of 10,000 columns, so any operation we do to the data will cost time. C/C++ is objectively faster than python or java. In addition, we’re the most familiar with the way data structures handle memory and store data in C/C++, making it the best choice.

**Technologies**

* Github
  + A technology that utilizes the git framework and allows for version control of software. Being able to ‘push’ and ‘pull’ projects files creates a cohesive structure teams can use in a development cycle.
* Visual Studio C/C++
  + As a technology this system shows the capabilities of the languages with which it is compatible. Meaning that for a program which is written in javaScript and is used to write C,C++,C#, and of course Java as well as many others.
* Mychron 5
  + The Mychron 5 is the data gathering tool that we will be using. The problem occurs with the software that Mychron uses, which was designed for users racing go karts. The customer who contacted us races outboard hydroplanes (boats) and is interested in getting the data processed in a different fashion. Because of this we will be using the Mychron 5 hardware and writing our own software to handle the data.
  + The device communicates over the CAN bus protocol. FTDI is a protocol used in translation.

**Process Model: Waterfall Model**

For our project we have decided to use the waterfall process framework for multiple reasons. The primary reason we are choosing the waterfall model is because it fits the strengths of the project and its weaknesses do not affect the project significantly. For example, the pros of the Waterfall model make the project ordered, principled, and easy to use. This is a great fit for us in the project and we are following along in class and learning in a linear fashion similar to the model. This should help us stay on track and highlight what we learn in class in the form of this project. In addition, the other pro is the Waterfall models ease of use which will continue to help us understand the key concepts involved in the class. While our project fits the model’s pros it also does not fit the cons (in a good way). We do not plan to deal with customer feedback and while the model might be rigid and hard to accommodate changes, there is not a much-expected change with our project. All we are interested in is the data supplied by the Mychron 5 program and what it can tell us about the hydroplanes. All in all, the waterfall model seems to fit very well for our project. Our project is fairly simple with clear requirements, in addition we do not expect to make many changes along the way, so the waterfall makes the most sense for us to use. Finally, as for why other models were not considered first, I’d like to bring up prototyping models. We do not expect to produce a prototype for this project which made it easy to take those models off of the board. In addition, evolutionary models were also not chosen because of the lack of expected change in our project’s life cycle.