oFreq Deliverables Schedule

~~**4/9/2013** – Planned directory structure for program executable files and libraries. This will be the organization of the actual compiled files. The directory structure should be planned to anticipate future growth and inclusion of the rest of the applications for the OpenSEA suite.~~

~~**4/9/2013** - Planned directory and file structure for single instance of an oFreq workflow. These are the working input files that a user will create to interact with oFreq and the output files generated by the program.~~

~~**4/20/2013** – First Iteration of library designed to support text file input/output with generic parameters. All Input files not complete.~~

~~**4/22/2013** – Second Iteration of library designed to support text file input/output with generic parameters. All besides forces.in is complete.~~

~~**4/24/2013** – Third Iteration of library designed to support text file input/output with generic parameters. All input files will be able to be read in properly.~~

~~**5/5/ 2013** – Definition of a class for a body will be implemented. This will include all user defined forces, hydro forces, and body momentum contained in the class definitions.~~

~~**5/9/2013** - Library to construct linear matrix from equations of motion and solve the equations by taking inverse of matrix and pre-multiplying to matrix of constants. This produces a solution to the equations of motion and describes the vessel motions.~~

~~**5/20/2013** - Library to define a generic set of equations of motion. This library may be pre-compiled and utilize some form of user input, or it may require user to compile a custom executable. Developer shall provide recommendation on the best method to achieve this goal.~~

5/25/2013 - Library to create derived set of outputs from basic output of object motions calculated from the library to construct linear matrix equations.

5/27/2013 – File Writer Class is implemented and complete. The File Writer Class and supporting classes will display the solution for each body into txt files which will be located in separate directories

5/29/2013 – Final Iteration of library designed to support text file input/output with generic parameters. All input files will be able to be read in properly and have been tested against errors. Runtime updates and errors will be logged in files located in the directory.

6/31/2013 – Debug/Log file support will be added. These will be two files in oFreq directory which will display current progress for log file and errors that may have abruptly ended the program in error file.

6/2/2013 - Library to perform parametric interpolation between data sets and between individual data points within a set. Output of interpolations will be with the library to construct linear matrix equations of motion. This includes support for the Hydro dynamic Database inputs. This deliverable is a lot of work and may need to be scrapped for final release, will re-evaluate once all other deliverables are complete.

6/2/2013 – Testing complete for final deliverables to ensure minimum errors.

6/4/2013 – Completed Doxygen documentation for future developers for this open source project. The doxygen documentation will include the UML classes.

6/5/2013 – Final Deliverables will be uploaded to sponsor website and other open source websites such as source forge and github.

6/6/2013 – Everything must be complete here. The date for my CSS 497 Project Presentation is 6/7/2013.

Possible Add-On features if time permits (Not required to this project)

- Library to construct linear matrix from equations of motion and solve for resonant frequencies. This requires the construction and solution of an Eigen-value problem. Output of Eigen-values and Eigen vectors is necessary.
- Library to construct derived input forces based on general user options and normal assumptions for marine vessels. Rather than the user explicitly defining how a force relates to the equations of motion, the application derives this relationship from more generic information.
- Support for the Windows OS.
- Multithread or Multiprocessor support if computation speed becomes an issue