Adam Kessler

Michael Riess

Ryan Hamilton

**Quad-Solver Testing Plan**

**Introduction:**

What follows is outline for testing the quad-solver program in the development process. A brief explanation is given along with specifics on who is to perform each test and where the results will be displayed or stored.

**Automation:**

Testing will be performed at compile-time for each unit, or function, and the program as a whole. The tests can otherwise be run using a shell script without the need of makefiles.

**Tests:**

* + System Testing:

A shell script will execute the compiled program. It will read inputs from a file, running the program with each set and testing the output against pre-validated quadratic equation solutions. The script will print to the console only those tests which were failed. Michael Riess will write the script, and Adam Kessler will be in charge of running the battery of tests once the program is completed and compiled.

* + Unit Testing:

Each function of the program is tested independently of the rest to ensure accuracy. C-unit tests, with asserts, are used as they are common and are relatively easy to write during the development process. A variety of inputs are tested to ensure that functions operate correctly e.g. using floats, nans, infinites, non-numbers, etc. to test input validation. Everyone is responsible for writing tests alongside the program code. Additionally, everyone is responsible for ensuring that all test clear before changes to code are pushed. Once the program is complete, a final runthrough of each test will be run by Ryan Hamilton. Just as with the system test, unit tests will only display the results of failed tests to the console.

* + Usability Testing:

Another team will be given access to the program and will provide feedback on its usability i.e. weather or not the software is easy to use, understand, etc. The team will be provided with a number of example problems that they will attempt to solve using the quad-solver program. Along with the direct user feedback, Michael Riess will monitor the test and make note of places where users are confused, make mistakes, etc.

Our team has agreed to perform usability testing on the other team’s project in return.

* + Standards Compliance:
    - *508*:

As a command line program, 508 compliance does not fall within the scope of the project. Therefore, no attempt at 508 compliance will be made, and no testing of compliance will be performed.

* + - *FERPA*:

While this program will be completed in an academic setting, no student documents or data will be processed or stored. Therefore, the project is not under FERPA guidelines and so compliance is moot.

* + - *HIPAA*:

The program does not, and will not, have access to medical documents or data of any kind. Therefore, the project is not under HIPAA guidelines and so compliance is moot.