Dr. J. Kapenga

CS 4900

**Stories Report**

Axel Solano, Gregory Smith

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stories** | **Time Estimation** | **Risk Estimation** | **% Completion** | **Actual**  **Time** | **Complete?** |
| **Version Control**  10/08/18  Greg   * Setup version control in github * Members are expected to push updates frequently | 1 hour | 1 | 100% | 1 min | YES |
| **Programming Standard**  10/15/18  Axel   * Our group has chose to comply with GNU’s C programming standard * Read formatting guidelines for C as defined in the standard | 1 day | 1 | 100% | 4 hours | YES |
| **C Unit Testing**  10/15/18  Axel   * Investigate & compare unit testing frameworks for C   10/22/18  Axel   * Agree on a C unit-testing framework * Read documentation for frameworks | 2 days | 2 | 100% | 2 day | YES |
| **Project Phase I**  10/22/18  Greg   * Language: gcc * Platform: Linux * Program Requirements:   + Quadratic equation solver for Dr. Kapenga - CS 4900   + Must accurately solve quadratic equations   + Input - Must take in 3 single precision (IEEE 32 normalized - float) numbers as arguments   + Output - Must output roots of quadratic equation * Phase I:   + Develop core program   + Use input args to output the quadratic equation   + Must comply with GNU standards   + Use double precision for calculations & output * To compile our c source file, use:   + gcc -std=c99 -o qSolver qSolver.c -lm     - -lm is needed or else gcc will not recognize the required math library   + This creates an executable file in the same directory called ‘qSolver’ * To run the executable, use:   + ./qSolver <value a> <value b> <value c>     - Values a, b, & c are the numbers that belong to the quadratic equation to be solved     - If arguments are not given, the user will be prompted to input these values | 3 hours | 1 | 100% | 1 hour | YES |
| **Implement CUnit Tests**  10/15/18  Axel   * Create unit tests for program * Testing plan:   + Input verification     - Check type of input - must not be Inf, Nan, or NA     - Show warning if input is out of range of single precision     - Check parameters of methods   + Output verification     - Check output of methods and program     - Verify correct answers given certain inputs   + Floating point precision     - Test ranges of inputs and outputs to match that of client’s needs | 7 hours | 1 | 100% | 9 hours | YES |
| **Project Phase II**  10/22/18  Greg   * Implement variable and parameter verifications * Implement warnings * Implement automated unit testing | 6 hours | 1 | 100% | 8 hours | YES |