**James Rehak**

AIT 642 / COSC 603

Software Testing and Maintenance

Spring 2016

**Project #1 – Reengineering Legacy Systems**

**Task 5 – Program Modification / Reengineering.**

I initially went into this project thinking it would be a piece of cake. The equations were written out with a diagram showing the flow of logic. I thought I wouldn’t even half to look at the Fortran code. I was very wrong! I learned that even “good” looking documentation does not necessarily mean that it is correct. There were enough inconsistencies and omissions to make it almost useless. My initially strategy of just writing the equations as methods and using Flow chart to decide when they were to be called in the execution did not work. I had to go through each line of code to double check everything.

The documentation did not explain how the drying factor was calculated. The program reverses the constant values a and b from the documentation, though in the Fortran equation to calculate Fine Fuel moisture a and b positions reversed as well, meaning that the documentation calculation is actually correct. It was confusing. There are also other discrepancies, such as the documentation says to change all spread indexes to 1 if the FFM is greater then 33%. In the code it is 30%. There were also some number formatting inconsistencies in the code. The author did not always write numbers to the 10th place or place a zero in front of the decimal. This hurt readability, especially because the code looked like it was typed on a typewriter and had ink splotches.

I found using git to be very helpful. I was able to work from multiple computers with no issues. I ended up not really using the branch feature that much due to the small size of the program and because I was working alone. I noticed my programming style getting sloppy once I started trying to correct all of the errors in my initial writing. This ended up costing me some time reformatting the code closer to the accepted guidelines.