Process Plan– Team 02

1 Introduction

The software will allow students to calculate the average number of words per sentence in a file. Using the command line, the student will call the software and specify a required file path to the file in which to calculate the average number of words per sentence. The user may specify optional parameters to specify which characters count as sentence delimiters and the minimum character length of a word. The program will read the specified file, perform the calculation and display the results before exiting.

2 Process Description

Below lists the set of activities needed to successfully perform the process; for each activity, the following are provided:

- Activity name (verb or verb phrase)
- Activity description (concise paragraph)
- Entrance criteria (inputs needed for the activity)
- Exit criteria (outputs produced by the activity and how you know it has been completed satisfactorily)

1. **Accept and parse the input file argument**
   The process must successfully accept and parse the given argument that specifies the file path and name. The input needed is provided in the args parameter to the main class. The output produced from this activity is a determination that the file specified by the user does or does not exist. If the file exists, continue. If the file does not exist, print a message and exit.
2. **Accept and parse the optional -h flag**
   The process must recognize when a -h flag has been input in the main class. The output produced from this activity is to display the help text and exit. All other options and arguments are ignored if they are specified.
3. **Accept and parse the optional -d flag**
   The process must recognize when a -d flag has been input in the main class. The output produced from this activity is to save the values that were specified for this flag as the *delimiters* that the user wishes the program to use in reading/determining a sentence termination. If this flag is not given, default the values of the *delimiters* to the following characters: . ? ;
   The exit criteria is that the program value for *delimiters* is set.
4. **Accept and parse the optional -l flag**
   The process must recognize when a -l flag has been input in the main class. The output produced from this activity is to save the values that were specified for this flag as the *minimum word length* that the user wishes the program to use when determining if a word should be included in the program calculation. If this flag is not given, default the value of the *minimum word length* to 4 characters.

5. **Calculate the average sentence length**
   The process will open the specified input file and read each sentence. The input to this process is the filename given by the user. The exit criteria is when EOF (end of file) is reached and average sentence length is computed.
6. **Display the output**
   The process will display the average sentence length to the user. The input for this process is the calculated average sentence length in the given file. The exit criteria is reached when the process displays this information to the user.

3 Team

The team consists of Reece Karge, Theodore Korolchuk, Kevin Stone and Traci Fairchild.

On this project there are four main roles:

1. Project manager - Organizes and manages the clients, requirements, project team, deliverables and other issues throughout the scope of the project.
2. Development lead - Lead developer in a team of developers, or working individually, who manages the development of a coherent system that is relevant, functional and of high quality according to the requirements.
3. Documentation lead - Responsible for creation of relevant documentation for the project
4. QA manager - Responsible for creation of test plans as well as development and execution of test cases

Role	**Team Member**
Project Manager	Reece Karge / reecekarge@gmail.com
Development Lead	Theodore Korolchuk / ted81@gmail.com
Documentation Lead	Traci Fairchild / tfairchild3@gatech.edu
QA Manager	Kevin Stone / kevinleestone@gmail.com

4 Estimates

Provide estimates for the following metrics:

- Effort hours: 40 team member hours
- Lines of code: 200-300 lines of code expected