Team 5 Testing Plan

Testing Process

Each method will be isolated in its class and built along with any dependencies necessary so that a main driver can be run from the command line, along with a file containing any necessary inputs. The expected output will be compared to the received output to determine success or failure.

Requirements Traceability

Many of the methods being used will have a dependency on other objects, classes, or methods in other directories existing in the Miradi repo. These will also be tested and well-documented to ensure that there are no loose-ends to minimize potential reasons for a failed test.

Schedule

Testing schedule as follows:

8-13: Revised testing plan presented, 5 test cases specified

8-15: All teams have success building classes from command line

8-20: Final decisions of classes/methods to test

8-27: Implemented at least one test using automated testing script

9-5: **Deliverable #3 -->** Automated testing framework designed and operable

Also, at least 15 test cases designed (and hopefully tested)

9-12: At least 20 test cases designed (and hopefully tested)

Also, begin “testing” testing framework

9-17: **Deliverable #4 -->** Testing framework revised and finalized, 25 test cases designed and tested

9-20: Re-analysis of testing framework and test cases, inspection for possible faults

9-24: **Deliverable #5 -->** Fault injection testing & report

9-27: Final report accumulation, and presentation designed

9-1: Team comfortable with project presentation; have practiced more than once

12-3: **Final Report -->** Project presentations

Procedures

Each method will be traced and recorded, ensuring that any necessary inherited classes or abstract methods are appropriately documented. From this information, we will study what is being performed so that we can understand the purpose of the function as well as predict the overall outcome. Test cases will be developed and written down, along with their expected outputs. After testing, pass/fail will be listed as well as any reasons why the test case was successful or a failure.

Hardware and Software Requirements

In order to enact our system of testing we will be using a Linux-based VM terminal as well as an installed JDK and JRE. This can be performed by entering '$ sudo dnf install java-latest-openjdk' on the command line.

Constraints

Our current and predicted future constraints are as follows but not limited to:

- Lack of prior knowledge and experience in github forking/cloning, as well as Linux
- Omission of an easy-to-use build file provided by Miradi
- Poor communication and group understanding placed by virtual constraints due to COVID
- Time management issues throughout the semester colliding with other courses
- Having to retrace certain steps as we encounter dead-ends in our progress

System Tests

The first 5 test cases have been uploaded to our team wiki, with test files created for each method. Methods are described as where they are in the code, as well as all dependencies the require in order to be run.