Team 6 - CSCI 362
Alan Arsiniega, Cameron Reuschel, Jacob Ballou
Project 1 Documentation

## Chapter 1

Introduction

In this stage of the project, our team has compiled the OpenMRS software and has run the included snapshot tests. The goal of this stage is to run the testing portion of the software, confirm that it passes its built-in tests so that it will be stable for future custom testing. Below are the steps we went through to prepare, compile, and test the program.

Preparation for the Deployment of OpenMRS

For the compilation and deployment of OpenMRS on Ubuntu, a few programs were required. A version of Java 8 was required for the program to function. For this task, we installed OpenJDK8, as it provided us with the level of functionality we needed for the project.

Git is also required for pulling the repository on github. There are multiple versions and extensions for OpenMRS. We pulled the core version, as this was the only part of OpenMRS 2.3.0 pertinent to our use of the webapp.

For the compilation of OpenMRS, the program required Apache Maven, a software management tool. For this project, the latest version of Maven, 3.6.0, was utilized. This allowed us to successfully build the program and launch the webapp.

Compiling of OpenMRS

The repository was pulled from Github and compiled using Maven. Depending on the machine and how it handled failures in testing, the compilation took between 2 to 20 minutes. Upon the first compilation, Maven initiated tests immediately afterwards, giving off errors within the openmrs-api component of the software. After not finding an immediate solution, it was decided to recompile the software, and retest it after its completion.

Testing of OpenMRS

Several tests were run using the provided test cases within OpenMRS. Through the use of Maven, the tests were iterated for OpenMRS, and each of its components. The tests were broken into 6 categories:

- OpenMRS
- openmrs-tools
- openmrs-test
- openmrs-api

- openmrs-web
- openmrs-webapp

Upon the testing of the most recent release of OpenMRS core, version 2.3.0-alpha, there were some test failures experienced during the run. Maven encountered two failures while testing the openmrs-api, and refused to test further.

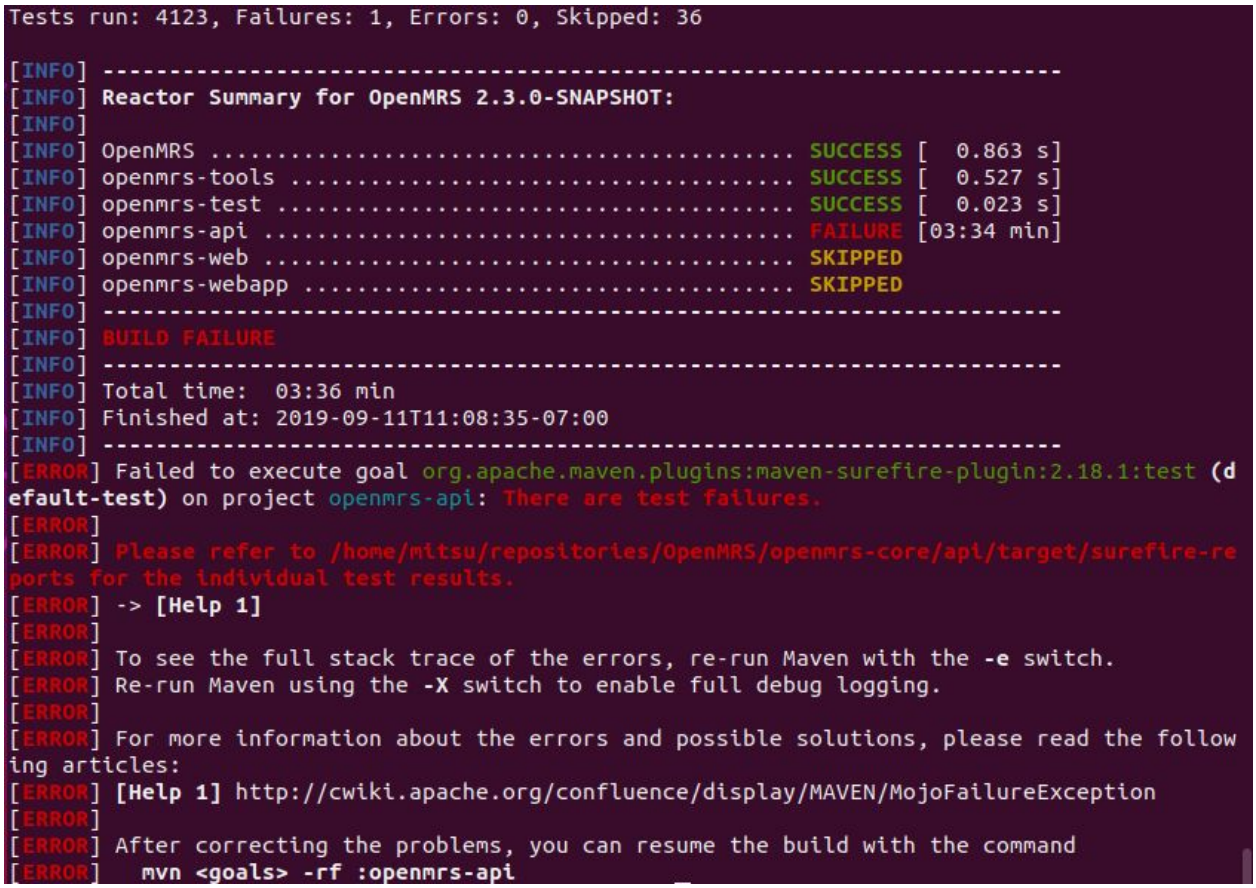

Figure 1: Test Run of OpenMRS core 2.3.0 with Maven. The test was unsuccessful

Resetting to last working version

Every decent version control system should be able to revert back to any point of the software's development life cycle. Because the code has been tracked by Github since 2012 we were able to see that the release being compiled had just been merged to master on August 29, 2019. By issuing the git reset command to the last working version, 2.2.0 from March 21, 2019, we were able to run the tests without so many errors. In fact, instead of skipping entire modules of the OpenMRS project like the web and webapp modules, we were able to successfully execute all tests in 11 minutes as shown below.

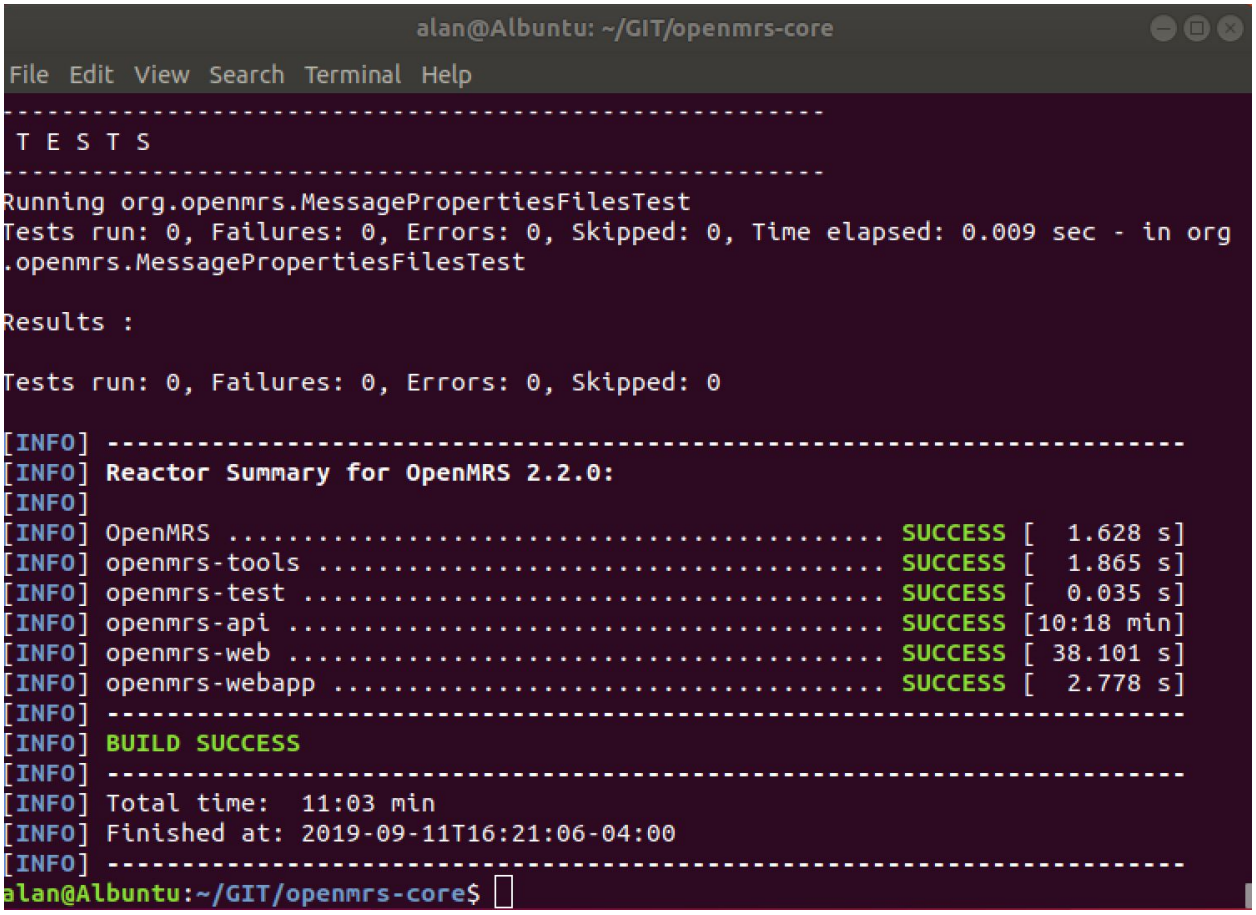


Figure 2: Test Run of OpenMRS core 2.2.0 with Maven. The tests are successful

## Conclusion

The installation of OpenMRS and testing has been more-or-less a straightforward process. Because there are currently some issues with the newest version of OpenMRS we have opted to continue working on the last known working version which happens to be 2.2.0. This is also the reason many companies choose not to upgrade to the latest and greatest systems, sometimes in a rush to release newer versions, not all the bugs and errors are really captured and understood.