## Chapter 3

Experience:

Deliverable #3 was a challenging yet rewarding experience. Further experience was gained in the bash scripting language. One significant challenge, which consisted of difficulty in discovering a bug in otherwise seemingly functioning code, also presented itself within the bash scripting language. For example, the getExpectedResult() function would loop through the lines of each test case text file in order to retrieve the expected output of the function to be tested based on a specified input. This method functioned as intended for the first test case file, but instead returned what appeared to be an empty or wordless string for the other four test case files. This was puzzling, as all test case files are structured identically. After research regarding this problem, it was discovered that the bug was due to hidden, incorrect end-of-line characters caused by differences in various text editors. Once this cause was discovered, simply two lines of code per variable corrected the issue.

Another challenge that we faced as a team for this particular deliverable was working together on code virtually and in real time because this deliverable consists of significant more scripting than past deliverables. Therefore, we decided it was best to split most of the work up between each group member. For example, each group member was responsible for specific functions of the script. The need to review and understand the work of others in order for another group member to add their assigned portion of work to the script was also somewhat of a challenge. This emphasized the importance of commenting code so that others are able to more easily understand another's work.

Despite these challenges, however, we seem to be working together well as a group thus far. We recognize that we work best when communication is frequent and clear between group members. This allows for further understanding of our project as a whole for every member of the group, especially when working on a project without the ability to meet in-person.

Automated Testing Framework Architecture:

The framework of our task runs on an Ubuntu virtual machine and is invoked by the runAllTests.sh bash script. The script exists within the directory, scripts, which is within the top-level directory, testAutomation. The necessary meta-data for the framework, such as input data and expected output for each test case, is found within each of the test case text files. The contents of each test case file conform to a specifications template. The bash script makes use of each test case file to locate the code to be tested, compile and execute the code with the specified input, and collect and compare the actual results to expected results of the test case. Once the script is run, each test case is performed and an html output file containing the results of each test case is opened and displayed within a browser.

How-To Documentation:

All necessary directories and files are accessible within the team GitHub repository. Running the script will automatically run all test cases and document testing results within a test report output file, named out.html. After accessing and cloning the Team-6 repository into a working directory on a Unix box:

1. Navigate by the command line/terminal prompt to the top-level directory, testAutomation: cd Team-6/testAutomation

2. Run the script: ./scripts/runAllTests.sh

First 5 Test Cases:

The first five test cases for our project test the contains() method of the OpenMRS code within the ThreadSafeCircularFifoQueue.java file, and each case appears as follows within its individual test case text file.

Test Case ID: 1
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/util/ThreadSafeCircularFifoQueue.java
Method Being Tested: contains
Requirement: Determine elements in a queue to ensure security and reliability.
Test Case Driver: ../testCasesExecutables/testCase1Driver.java
Test Data: null
Expected Result: false

Test Case ID: 2
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/util/ThreadSafeCircularFifoQueue.java
Method Being Tested: contains
Requirement: Determine elements in a queue to ensure security and reliability.
Test Case Driver: ../testCasesExecutables/testCase2Driver.java
Test Data: 43
Expected Result: false

Test Case ID: 3
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/util/ThreadSafeCircularFifoQueue.java

Method Being Tested: contains
Requirement: Determine elements in a queue to ensure security and reliability.
Test Case Driver: ../testCasesExecutables/testCase3Driver.java
Test Data: 2344
Expected Result: false

Test Case ID: 4
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/util/ThreadSafeCircularFifoQueue.java
Method Being Tested: contains
Requirement: Determine elements in a queue to ensure security and reliability.
Test Case Driver: ../testCasesExecutables/testCase4Driver.java
Test Data: foo
Expected Result: false

Test Case ID: 5
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/util/ThreadSafeCircularFifoQueue.java
Method Being Tested: contains
Requirement: Determine elements in a queue to ensure security and reliability.
Test Case Driver: ../testCasesExecutables/testCase5Driver.java
Test Data: barr
Expected Result: false