CSCI 362 Fall 2017
Software Engineering
Bowring

Team Term Project Specifications

Project:

Design and build an automated testing framework that you will use to implement your test plan for your chosen software project from **Deliverable #2**. The testing framework specifications:

Introduction:

The testing framework will run on Ubuntu (Linux/Unix). The testing framework will be invoked by a single script from within the top level folder using "./scripts/runAllTests.(some scripting extension)" and will access a folder of test case specifications, which will contain a single test case specification file for each test case. Each of these files will conform to a test case specification template that you develop based on the example template below. Each test case specification file contains the meta-data that your framework needs to setup and execute the test case and to collect the results of the test case execution.

Framework Directory Structure:

Create all these folders even if you do not use them; create additional folders as needed. This structure will exist in your GitHub team repository.

Note: **No Spaces** in any folder or file names

```
/TestAutomation
    /project
        /src
        /bin
        / …
    /scripts
        runAllTests.(some scripting extension)
        ... other helper scripts
    /testCases
        testCase1.txt
        testCase2.txt
        …
    /testCasesExecutables
        testCase1(may be folder or file)
        ...
    /temp (for output from running tests … be sure to clean at start of runAllTests)
        testCase1results (might be folder or file)
        ...
    /oracles
        testCase1Oracle (might be folder or file)
        ...
    /docs
        README.txt
    /reports
        testReport.(txt or html)
```

Test case specifications template example:

Create a test case template in the form of a .txt file that you will use to specify each individual test case. This template will provide for comments in addition to a mechanism to specify the following test case properties as a minimum set. All folder and file references within a test case specification must be relative not absolute.

1. test number or ID
2. requirement being tested
3. component being tested
4. method being tested
5. test input(s) including command-line argument(s)
6. expected outcome(s)

Testing Report:

Each time the framework is invoked, it will produce and automatically open a professional-grade testing report detailing the test cases and the results of each test case execution.

runAllTests.[some scripting extension]:

1. From within the top-level folder ("TestAutomation"), the command to start your framework must be "./scripts/**runAllTests**.(some scripting extension)"
2. **runAllTests** will walk the folder /testCases and use each test case specification file found therein to instantiate and execute a single test case; as an example, the script will use the test case to locate the code to be tested, then compile it, then execute it with the specified inputs, collect the results, compare the results with the expected results, and save the relevant facts for inclusion in the test report
3. **runAllTests** must open and display a Testing Report in a web browser using html
4. executing **runAllTests** repeatedly on the same code base should not change its behavior or its results

Evaluation:

I will clone your team directory from GitHub into a working directory on my Unix box. I will read the README file and follow its instructions for dependencies, etc. if any. Then I will navigate to the top-level directory as specified above and execute "./scripts/**runAllTests**.[some scripting extension]". Be sure you try this yourself in a clean location. One common error is that there exist absolute path references in the script that break the framework. The script should launch a browser with your testing report. If the script does not execute, you will have **one** opportunity to repair it.

Note:

This project is difficult and time-consuming. Planning is critical. Read and follow the instructions – this is software engineering after all. Procrastination will cause failure. Please consider me a resource that you can consult at any time – sometimes students like to think of me as a customer for this project. I encourage you to collaborate, to share, to explore, and to have fun!