

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 per the test case specification template below. Each test case specification file contains the meta-data that your framework needs to setup and execute the test case and 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.

/[Team name in [CamelCase](#)]_TESTING (Note: No Spaces in any folder or file names)

 /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:

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 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, 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
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