CSIS0403/COMP3403
Implementation, Testing, and Maintenance of Software Systems
Department of Computer Science
The University of Hong Kong

Assignment 1 – Practical Black Box Testing and Test Automation

Part 2

Description

After our pause for the quiz we can now return to Assignment 1. In this short continuation you will perform the actual testing. Essentially it involves simply running test cases and random tests using your test harness and documenting the results. You will:

- execute the structured test cases you developed in Part 1;
- perform a cycle of random testing;
- based on your results, draw conclusions about the relative effectiveness of your structured testing and random testing.

This exercise should give you some insight into the quality of your original test set and the limitations of each approach. We have seeded the IUT with various defect regions, including several boundary defects.

Tasks

As described in Part 1, you will work with a JAR file named `IndexEngine.jar` containing a package `C3403.A1`. In the package are an interface, `IndexEngine`, and two classes that implement that interface: `IUTEngine` and `OracleEngine`, representing the IUT and oracle implementations respectively.

You can consider the oracle to be an old and inefficient implementation. **It runs very slowly.** This is a factor that can work against random testing as the sole black box technique used to test an application since it imposes a practical limit on the number of random cases that can be executed within time and other resource constraints.

1. Execute your test cases:

a) Using the test harness and input file developed in Part 1, execute the test cases you specified in Task 1 of that assignment. **Do not modify your cases**;

b) Log the execution: See Appendix 1;

c) Prepare a brief Anomaly Report for any failure you observe: See Appendix 2;

The Appendices to this assignment sheet contain the relevant extracts from IEEE Std 829-2008.

2. Perform random testing on the IUT:

a) Perform a single cycle of random testing with 10,000 randomly-generated test cases. Test without using any knowledge gained from results obtained in the previous Task. This is necessary because we wish to compare the two strategies;

b) Prepare a Test Report covering your random testing (See Appendix 3). List failed cases, if any, in a table as part of this report. **Do not write individual Anomaly Reports for random testing.**

Note: *there are no marks awarded for the number of defects you detect. You are random testing with a limited number of cases and you will certainly fail to detect several of the seeded defects.*

3. Compare the two approaches:

a) Your testing in Tasks 1 and 2 will reveal different types of defects. Compare the types of defects detected and, on that basis, write a short report (around 0.5 page) discussing the strengths and weaknesses of each approach. If you wish you may do some informal exploration with additional cases if it will help you develop your answer.

Deliverables

Task 1.
1. Test Log;
2. Anomaly Report for each failure (if any);
3. The file containing your Decision Table/Streamlined Domain Testing cases from Part 1.

Task 2.
4. Test Report;

Task 3.
5. Report comparing the strengths and weaknesses of the two test strategies.

Deadline

Thursday April 7, 23:55

Appendix 1. Test Log Outline

1) Document identifier

2) Details of the Level Test Log

- Description

  Provide any general information that applies to all entries in the log (exceptions can be specifically noted in a log entry). The following information may be considered:
  - Identify the items being tested including their version/revision levels
  - Identify any changes from the prior testing documents to the attributes of the environments in which the testing is conducted
  - Date and time of start and stop
  - Name of the individual running the test
  - Any issue that causes testing to halt

- Procedure results

  For each execution *(you may have only one if you did not encounter any difficulty in running your test cases from file)*, create a record of the results (*manually or automated as part of your harness*). Record the success or failure of each test case *(show only test case id and result)*.

- Anomalous events during test execution (other than test case failures)

  Record what happened before and after an unexpected event occurred. Record all circumstances surrounding the inability to begin execution of a Test Procedure or failure to complete a Test Procedure. If this information is recorded in an Anomaly Report, simply reference the Document identifier.

Appendix 2. Anomaly Report Outline

For each anomaly, include any of the following information that is relevant (anomalies include failures in the test procedures, scripts, etc., as well as test failures in the IUT).

1) Document identifier

2) Description of the anomaly

Provide a description of the anomaly. Indicate whether the anomaly is reproducible, and provide enough information to make it reproducible if it is. This description may include the following items *(the first three are usually sufficient for IUT failures)*:

- Inputs
- Expected results
- Actual results
- Unexpected outcomes
- Procedure step
- Environment
- Attempts to repeat

Related activities and observations that may help to isolate and correct the cause of the anomaly may be included (e.g., describe any test case executions that might have a bearing on this particular anomaly and any variations from your Test Procedure).

Appendix 3. Test Report Outline

1) Document identifier

2) Detailed test results *(only for the random testing task, and with details only for failed cases.)*

- Summarize the results of testing and identify all unresolved anomalies other than test case failures.

- Report any variances of the test items from their specifications *(these are the failed cases - use a table format)*. Indicate and explain variances from the test documentation (e.g., test procedure changes or tests not executed).