

Asg 4 - Debugging

Value: 20%

Due date: 16-Oct-2015

Return date: 06-Nov-2015

Submission method options

EASTS (online)

Alternative submission method

Task

Assignment 4 revolves around debugging a small application.

You will be provided with some code which contains several known bugs. The source code and initial bug reports will be made available in the Assignment 3 directory of the Resources section of the subject's Interact site.

Your task is to replicate, simplify, trace, and eliminate these bugs using best practice debugging techniques. Your debugging effort should take place within a version control system using appropriately sized, commented, and timely commits.

Please note that the focus in this assignment is on the process and techniques applied, not the complexity of the code. It is recognised that the process applied is 'way over the top' for the size of problem addressed.

Tasks

1. Replication. (20%)

- a. For each of the reported bugs, generate a test case in UAT format that will reliably reproduce the buggy behaviour.
- b. Commit the test case documentation to version control.

2. Simplification. (20%)

- a. For each of the reported bugs, produce a simplified and automated test that reliably reproduces the bug.
- b. Commit the test code demonstrating the bug to version control.
- c. Record output from your test, demonstrating the buggy behaviour.

3. Tracing. (30%)

- a. For each of the reported bugs, trace back through the programs dependences from the point of failure to find the origin of the bug. Use a sequence of hypotheses and tests to support this search, and document this sequence in a debugging log.
- b. For each of the reported bugs, demonstrate that before the identified origin, variable values are 'sane' and that after the origin, at least one value is 'infected'.
- c. Record screen shots demonstrating the origin of the bug.

4. Resolution. (20%)

- a. For each of the reported bugs, correct the bug.
- b. Commit the corrected code to version control.
- c. Record output listings from your automated test demonstrating that the buggy behaviour has been eliminated.
- d. Carry out the user test case with the specified data and show that the buggy behaviour has been eliminated.

5. Version Control (10%)

- a. Carry out each commit to your VCS as requested above.

Submit:

A word processed file containing:

- a. The URL for repository used for your debugging effort. Make sure you commit both at the start of the debugging process, once you have developed an automated test demonstrating each bug, and after you have resolved each bug. Include a directory at the top of the file indicating where in your repository the files related to each debugging effort can be found.
- b. For each bug:
 - i. The user tests (in UAT format) that demonstrate the bug. (Specifying expected buggy output)
 - ii. Output from your automated test demonstrating the buggy behaviour.
 - iii. A debugging log recording the succession of hypotheses, tests, and results of those tests that you used to trace the origin of each bug.
 - iv. 'Before' and 'after' screen shots identifying the origin of the bug. (Show variable values)
 - v. Output from the automated test demonstrating correct operation after resolution of each bug.
 - vi. Test results from the user test you specified for each bug demonstrating correct operation.

Rationale

This assignment requires student to demonstrate skills related to the following learning objectives:

- apply specialised and technical knowledge and skills to professional programming practice;
- be able to critically analyse software programs and apply theories of debugging to complex debugging problems;

Marking criteria

Replication (15%)

Criteria	Fail	Pass	Credit	Distinction	High Distinction
How effectively has the replication process been applied?	Generally described by the following characteristics: Description of circumstances and actions leading to expression of bug missing or incorrect. Description of test procedure missing or incorrect. No test data. No supporting data specified.	Generally described by the following characteristics: Minimal and vague description of circumstances and actions leading to expression of bug. Minimal and vague description of test procedure. Generalized test data Some supporting data not specified.	Generally described by the following characteristics: Minimal description of circumstances and actions leading to expression of bug. Minimal description of test procedure. Generalized test data. Supporting data specified	Generally described by the following characteristics: Thorough description of circumstances and actions leading to expression of bug. Thorough description of test procedure Specific test data Generalized supporting data specified	Generally described by the following characteristics: Detailed and comprehensive description of circumstances and actions leading to expression of bug. Detailed and comprehensive description of test procedure Specific test data Specific supporting data specified

Simplification (20%)

Criteria	Fail	Pass	Credit	Distinction	High Distinction
How effectively has the automation and simplification process been applied?	Generally described by the following characteristics: Test does not reliably demonstrate bug. Test not automated. No test harness. No simplification.	Generally described by the following characteristics: Test reliably demonstrates bug Test requires interaction Test supported by customized 'test harness' Automated test shows no simplification from initial test case.	Generally described by the following characteristics: Test reliably demonstrates bug. Test does not require interaction. Test supported by customized 'test harness'. Automated test slightly simplified from initial test case.	Generally described by the following characteristics: Test reliably demonstrates bug Test does not require interaction Test integrated with unit test framework. Automated test considerably simplified from initial test case.	Generally described by the following characteristics: Test reliably demonstrates bug Test does not require interaction Test integrated with unit test framework. Test simplified to smallest subset of input and complete code modules that demonstrate bug.

Tracing (35%)

Criteria	Fail	Pass	Credit	Distinction	High Distinction
Does the debugging log demonstrate an effective use of scientific debugging procedure?	Generally described by the following characteristics: Incorrect origin identified. No hypotheses and tests recorded. No debugging process shown. Transition from sane to infected not demonstrated.	Generally described by the following characteristics: Correct origin identified. Minimal hypotheses and tests recorded. Debugging process based on intuition, no logical progression shown. No clear demonstration of transition from sane to infected.	Generally described by the following characteristics: Correct origin identified. Minimal hypotheses and tests recorded. Debugging process based on intuition, some logical progression shown. Clear demonstration of transition from sane to infected.	Generally described by the following characteristics: Correct origin identified. Substantial set of hypotheses and tests recorded. Debugging process based on deduction and observation, good logical progression shown. Clear demonstration of transition from sane to infected.	Generally described by the following characteristics: Correct origin identified. Complete set of hypotheses and tests recorded. Debugging process based on deduction and observation, rigorous, step by step logical progression shown. Clear demonstration of transition from sane to infected.

Resolution (20%)

Criteria	Fail	Pass	Credit	Distinction	High Distinction
Has the bug been correctly remedied?	Generally described by the following characteristics: No	Generally described by the following characteristics: Identified bug resolved. Bug fix introduces other defects. Test harness confirms initial defect resolved. UAT confirms initial defect resolved, but may show new issues. No regression testing.	Generally described by the following characteristics: Identified bug resolved. Bug fix introduces no other defects. Test harness confirms initial defect resolved. UAT confirms initial defect resolved. No regression testing.	Generally described by the following characteristics: Identified bug resolved. Bug fix introduces no other defects. Integrated and automated test confirms initial defect resolved. UAT confirms initial defect resolved. No regression testing.	Generally described by the following characteristics: Identified bug resolved. Bug fix introduces no other defects. Integrated and automated test confirms initial defect resolved. UAT confirms initial defect resolved. Regression testing confirms no new defects.

Version Control (10%)

Criteria	Fail	Pass	Credit	Distinction	High Distinction
Were the VCS commits specified in the assignment instructions carried out at the appropriate times?	No evidence of version control.	All tests, documentation and code changes committed after all bugs resolved.	Tests, documentation and code changes committed after each bug resolved.	Test case UAT committed before debugging. Automated tests and code changes committed after each bug resolved.	All commits made as requested.

Presentation

There are two aspects to the presentation of this assignment.

The first is a word processed file submitted through EASTS which contains:

- The URL for the version control repository where all code development and tests for the debugging assignment can be found.
- for each bug:
 - the user test case for each bug
 - the output from the automated test demonstrating the buggy behaviour
 - your debugging log showing the sequence of hypotheses and tests leading to identification of the origin of the bug
 - screen shots demonstrating the transition from sane to infected at the origin
 - output showing resolution of the bug

The second aspect of the assignment is presented as a functional code repository containing the complete development history and code for the assignment. You should clearly comment the update which represents the point at which you submit the assignment.

It is critical that you do not delete the repository until after you have received your mark for the assignment.