CS 1632 - Software Quality Assurance

Spring Semester 2019

\* ASSIGNED: 14 JAN 2019 (MW), 15 JAN 2019 (TH)

\* DUE: 4 FEB 2019 (MW), 5 FEB 2019 (TH)

\* LATE DUE DATE (-15): 6 FEB 2019 (MW), 7 FEB 2019 (TH)

**Deliverable 1**

For this assignment, your group will determine a test plan for JBefunge, based on the requirements listed in the file requirements.md. Note that these are a \_subset\_ of all of the requirements of the program. There are several known defects in the software; you will need to find and report on at least three, as well as request one enhancement. Additionally, a traceability matrix showing the mapping of test cases to requirements is required.

The JBefunge IDE repository is here: https://github.com/laboon/JBefunge

Time will be given in class to group yourself into groups. There should be two and only two members of a group.

There should be, at an absolute bare minimum, twice as many test cases as requirements, although I would normally expect several test cases per requirement. Note that this DOES not mean that having exactly two test cases is necessary. It also does not mean that you should endeavor to have the bare minimum number of test cases and expect a perfect score. Anything below this number of test cases will not be considered an adequate test plan for this assignment.

Conversely, however, if the average (mean) number of test cases is more than 4x the number of requirements, you are probably overtesting for our purposes. Your goal is to get sufficient test coverage of the program at a level necessary to determine quality, not to exhaustively test the program.

There are ten requirements. You should split this up into two sections of the test plan and LABEL each section with which team member worked on that section. Assuming there are no disparities in the quality of the test plans, you will receive a single grade for the group. If there are disparities, then each member of the group will get a different grade for the test plan. I recommend that you work together on developing the test plan, or at a minimum, double-checking your partner's work. It is up to you how you would like to split the requirements, but I do require you to label the different sections of the test plan with who did which part.

The defects, enhancements, and traceability matrix will be graded as a group. Depending on how you split up the requirements, each team member may find a different number of defects. You should work together to discuss the defects, which are the three most important that you found, and how you want to describe them.

Each requirement should have AT LEAST one test case associated with it, and each test should have EXACTLY ONE requirement associated with it. This can easily be checked via a traceability matrix (which you should also deliver). Note that some test cases may actually test several requirements. You should specify the one that the test case fits best - that is, what are you really trying to test with this test case?

Test cases should mention all necessary preconditions, execution steps, and postconditions. You do not need to separate out preconditions vs input values or postconditions vs output values for this test plan.

It is NOT necessary to make multiple test plans inside a test suite; it is enough for there to be one test plan.

I expect you to test three edge or corner cases as part of the test plan. These NEED to be marked in the description of the relevant test case in order to receive credit for them.

It is expected that you actually execute the test plan in order to find the defects, along with some exploratory testing to determine how the system works and where defects might lie. There are AT LEAST three defects. While you are not expected to find \*all\* of the defects, a reasonable test plan should definitely find at least three. This is an intentionally target-rich environment.

All defects should result in a failed test case (i.e., there should be three failing test cases). The defect should refer to the failing test case by name, number, or other identifier. If you find a defect via exploratory testing, you should go back and write a test case which causes the failure to ensure that it is fixed in the future.

HINT: Note that there should be no defects with the programming language itself, only the IDE. I do not expect you to learn yet another programming language for this course (Ruby will be quite enough!). All Befunge programs that I have added should work correctly. However, if you \*do\* find a defect in my implementation of Befunge, your group is entitled to bonus points on the assignment! Please include a SEPARATE page labeled BONUS with a list of any defects that you find related to the Befunge interpreter itself.

You will also write up one request for enhancement to the program. This should be something that is \_not\_ specified in the requirements, but you, as a budding Befunge programmer, would like to see added to the IDE.

## Format

Please hand in the paper to me with a cover page with:

\* The name of the project under test ("JBefunge")

\* The names of the people in the group

\* The title "CS 1632 - DELIVERABLE 1: Test Plan and Traceability Matrix"

There should be a short introduction (a few paragraphs) in which you may note any concerns or difficulties you may have had or anticipate with the testing process. You should also note why you considered certain test cases, how you thought of edge cases, etc.

This should be followed ON A NEW PAGE by the list of test cases. You may name or number them any way you wish, but be consistent. You should write them out in this format -

IDENTIFIER:

DESCRIPTION:

PRECONDITIONS:

EXECUTION STEPS:

POSTCONDITIONS:

The IDENTIFIER is some value which will UNIQUELY specify the test case. It should consist of a number or some other mnemonic (e.g. TEST-INVALID-TIMES, TEST-LOW-NUM-TIMES, etc.).

ON A SEPARATE PAGE, a traceability matrix should be provided mapping the test cases with their associated requirements. Remember that all requirements should map to AT LEAST ONE test case (but usually more), but all test cases should map to EXACTLY ONE requirement.

Finally, ON A SEPARATE PAGE, list three (and ONLY three) defects found. The defects should follow the defect reporting template:

SUMMARY:

DESCRIPTION:

REPRODUCTION STEPS:

EXPECTED BEHAVIOR:

OBSERVED BEHAVIOR:

SEVERITY:

IMPACT:

The test case which found the defect should be listed as part of the DESCRIPTION.

The SEVERITY and IMPACT will obviously be a bit subjective; use your best judgment and explain why you believe that

You may find it useful to review chapters 5 through 9 of the textbook.

## Grading

\* Introduction: 10% of grade

\* Test Plan: 50% of grade

\* Traceability Matrix: 10% of grade

\* Defects and Enhancements Found and Described: 30% of grade

The requirements are listed in the file requirements.md.

Please feel free to email me or come to office hours to discuss any problems you have.