

ASSIGNMENT: DOMAIN TESTING AND BUG REPORTING

Version 8
October 2011

In your risk-based testing lab, you listed a set of risks associated with a variable that you chose from Open Office Impress or Writer.

It is essential that you use a variable that is suitable for domain testing. For example, an enumerated variable is not suitable for domain testing. If you started working with a variable that is not suitable for domain testing, or if you have trouble analyzing the variable, I strongly recommend that you pick another variable and start over. An assignment that tries to kludge around a variable that isn't well suited to a domain analysis typically earns a grade of 10 to 40%

Conduct a domain analysis ON THAT ONE VARIABLE. Please conduct ALL of your tests on the same ONE variable.

Step-by-Step Specification of the Assignment

1. For your focus, pick one (1) input or output variable. Identify that variable and stick with it throughout testing. You will record your analysis in an introductory paragraph (see step 2), a table (step 3-11), and a summary paragraph (step 12).
2. List and briefly describe any output variables, storage-related variables, and variables that hold intermediate results of calculations that you can think of whose values depend on the input variable you are focusing on or that constrain the set of valid values for the input variable you are focusing on. If it's not absolutely obvious how these are dependent on your variable or how they constrain it, explain the relationships.

Students often forget to include this list and/or the summary. Please, give me a paragraph answer on each of these, even if it is short.

3. Identify risks (failure modes) associated with your input variable. A risk might involve a possibility of an error in handling the specific value(s) of the input variable or an error associated with some variable that is caused by the value of your input variable. (Or you might think of some other type of error--be creative.)

You probably have this list from the Risk-Based Testing Lab. If you restart with a new variable, I strongly recommend that you start by making this list, before designing any tests. You might have a very short list at first. Add entries to the list as you go. Work from the list of failures modes to create test cases.

4. For each failure mode, design one or more tests specifically directed to that risk. Each test will go on its own row on the table.

9. If you find bugs, check whether they are already reported bugs. If you find an old bug, write a brief description for me, cross-reference it to the OpenOffice bug number, add notes to the OpenOffice database record if you have something useful to add, but don't re-report the bug. If you find a new bug, write it up and report it into Open Office.
10. Run up to 15 tests. If you find one bug, subtract 5 tests (run 10). If you find two bugs, subtract 3 more tests (run 7). You must design and run at least 7 tests.
11. I strongly recommend that you pair with someone else to review bug reports before submitting them. I will review bug reports and downgrade your assignment if your reports are poorly written.
12. Write a summary report that explains what you now believe you know and don't know about the variable you tested. Justify this on the basis of your test results.

Please don't forget to write the summary report. This is another silly way that students lose points. They forget to write the summary. This is often the easiest part of the assignment. Please, collect your points on it.