

Black Box Software Testing

Domain Testing Assignment

Fall 2005 Assignment 2

This assignment is due on September 24, 2005.

Please use the latest version of Open Office 2.

OVERVIEW

- You are going to do a risk-based domain analysis of a function in Impress. If you aren't sure what that means, please review the lecture and slides.
- Select your function from the group of functions that allow you to create, edit, format a spreadsheet and position the spreadsheet on the page. These are the functions for establishing the look and feel of the spreadsheet, not for entering or analyzing data in the spreadsheet.
- Pick one function. (If two of you are working together, pick two functions.) Pick an input variable associated with that function. You will design all of your tests on the basis of this one variable of this one function.
- Your test design approach will be domain testing:
 - You will create equivalence classes of possible values of your input variable
 - You will focus on one of these classes, pick a best representative of that class, and base your test on that.
- The basis for separating values of your variable into different equivalence classes is risk – how the program could fail. A failure mode is a way that the program could fail. Any program can fail in many different ways. Each failure mode is a different way.
- You are looking for failure modes that are specifically related to the variable you are testing. For example, integer overflow might be a failure mode associated with a numeric input field that accepts integers.
- Once you have a potential failure in mind, consider the values of the variable under test that might trigger the failure—this is one equivalence class. Consider the values that could not trigger the failure. This is another equivalence class.
- When you think about failures, or about the results of your tests, don't just think about how the program will receive and filter your inputs. Also pay attention to the effects of your inputs, results of calculations based on your inputs, displays or graphics of your inputs or of values based on them, or how your data might be stored. Thinking this way can lead you to other interesting failure modes.

DETAILS

1. Pick one (1) input variable associated with the function you have chosen. **Name the function and the variable. Stick with the same variable throughout testing.**
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 who constrain the set of valid values for the input variable you are focusing on.
3. Design your tests. This includes:
 - 3a Identify risks (failure modes) associated with your input variable. A failure mode 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.)
 - 3b For each failure mode, divide the values of the variable you are working with into equivalence classes. At least one class should include values of your input variable that could trigger the failure. (If it's hard to think of your variable that way, work with another failure mode or start this assignment over with another variable.) Focus your attention on this equivalence class. Pick the best representative from this class—the value most likely to trigger a failure. Create your test using this value.
 - 3c Consider what makes this best representative a powerful test. In your table, you will explain what makes this best representative powerful. To write a good explanation, it might help you to compare your test to less powerful alternatives (other values of the variable you are studying).

- 4 Report your tests (all of your tests) and their results in a table format. Name the function and the variable in the table heading. The rows of the table show your tests, one row per test. **The table MUST include the following columns, IN THIS ORDER:**

Risk	What possible failure or type of failure do you have in mind?
Class of interest	What is the equivalence class of values (of the variable you are working) that <i>might</i> trigger the failure?
Test (best representative)	If the test is simply to set the variable to a given value, just list the value. If you are setting other variables too, list them and their values. If necessary, describe your sequence of actions.
Power	Explain what makes this test a powerful test <i>for the risk you have listed.</i>
Expected result	What result would have indicated a passing test?
Obtained result	What did you get?

5. If you find bugs, check whether they are already reported bugs.
 - If they are old bugs, cross-reference to them in your assignment, but don't report them in Issue Tracker.
 - If they are in the database as unconfirmed, report your replication in Issue Tracker.
 - If they are new bugs, write them up and report them into Issue Tracker.
 - If you don't do this (check for duplication and, if appropriate, file a bug report), I will not give you any credit of any kind for the bug.
6. Run up to 20 tests (up to 30 if you work with a partner, split equally across your two functions). If you find one bug, subtract 5 tests (run 15 or 25). If you find two bugs, subtract 10 tests (run 10 or 20). If you are working as a pair and find three bugs (at least one in each of your two functions), you can drop down to 15 tests, but there must be at least 7 tests per function.
7. I strongly recommend that you pair with someone else and that you carefully review each other's draft bug reports before submitting them. I will review bug reports and downgrade your assignment if your reports are poorly written.
8. Write a summary report that explains what you believe you now know and don't know about the function you tested. Justify this on the basis of your test results. If two of you tested two functions, write a summary report for each.