The Eleventh 2016 International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2016)

Simple spreadsheet test case application to test spreadsheet formula in end-user software engineering

Authors:
Noprianto
Benfano Soewito, Ph.D
Dr. Ford Lumban Gaol
Spits Warnars Harco Leslie Hendric, Ph.D

Presented by:
Benfano Soewito, Ph.D

Introduction

- In spreadsheet application, users may work with raw tabular data, which can be processed with formula and function.
- One formula can be applied to one or more input, but testing is often performed only for one input variant.
- If there is no error, and the result is considered correct, then the formula is applied to more data (the formula is considered valid).

The problem

- Real data may not provide all the possibilities to prove that one formula is correct.
  - Only when users can write test cases for many input variants, they can be sure that one formula is correct.
  - This is particularly true for complex formula definition.
- Writing test cases is not a simple task in spreadsheet, mainly because there is no standard way to do that.
  - Users may populate the data with some dummy or test data, but it will only test if there is any error when a formula is applied to all data.
  - There is no verification, compared with expected result.
  - May populate the worksheet with unneeded contents.

End-user software engineering

- Programming is defined as the process of planning or writing a program.
- Based on the goal, there are professional programmers (being paid to ship and maintain software) and end-user programmers (write program to support some goal in their own domain of expertise).
- End-user programming (create) → End-user development (create, modify, extend) → End-user software engineering (develop, operate, maintenance).

Spreadsheets

- They are interesting because they provide computational techniques that match users' tasks (that shield users from low-level details of programming) and their table-oriented interface (that serves as a model for application).
- They are also used to develop many serious applications.
- Spreadsheets are used in almost all businesses.

Testing and debugging in end-user software engineering

- Testing in end-user software engineering is usually overconfident (compared with cautious testing in professional software engineering).
- Debugging in end-user software engineering is usually opportunistic (compared with systematic debugging in professional software engineering).

Errors in spreadsheets

- There are logic error, reference error, placing numbers in a formula, copy/paste error, data input error and omission error.
- Errors can lead to poor decisions and even cost millions of dollars.

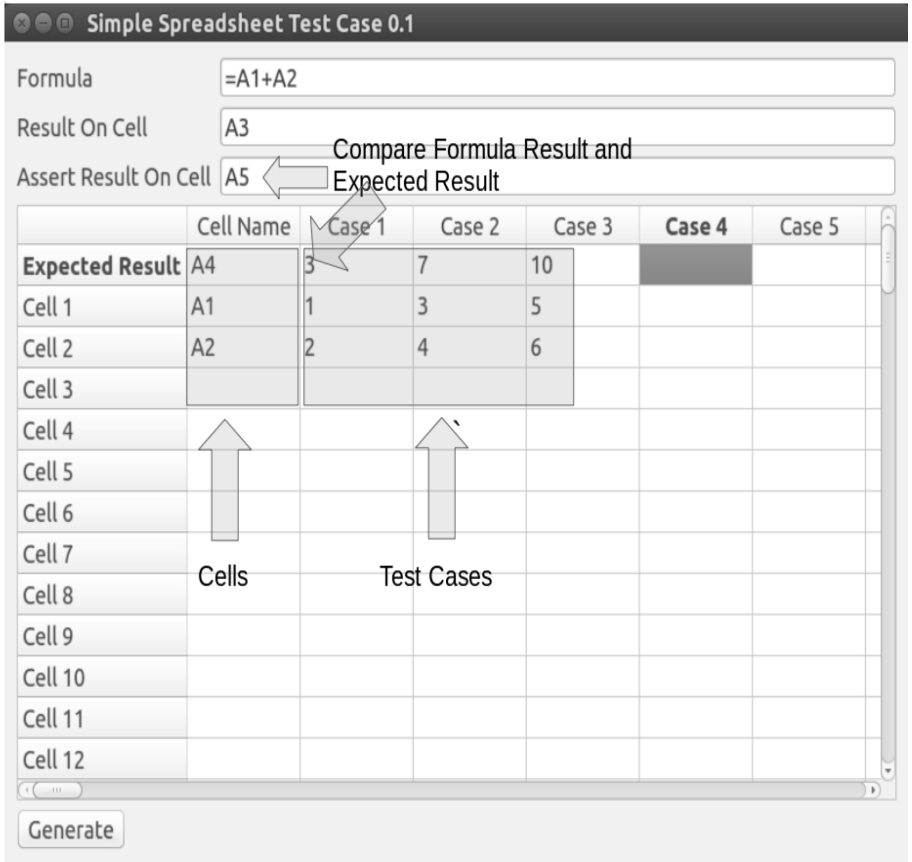
Proposed method

- Test case generation application
  - Standalone, multi-platform GUI application, does not require any spreadsheet software to be installed in the system.
  - The only needed inputs are the formula definition, where to put formula result, where to put comparison between formula and expected result, and one or more test cases along with expected result.
  - Output of this application is in Office Open XML (ISO/IEC 29500) spreadsheet, with .xlsx file name extension.
  - Number of sheets in this document will depend on number of test cases entered by user, plus one summary worksheet.
- Promotes test-driven development style (where test cases are written before the actual codes), but applied to spreadsheet.

How to use this application

- In the spreadsheet-like table interface, user can input cell names (which should be related with formula definition).
- Test cases, along with their expected results, may be specified, and will be read and processed, column by column, until empty cell is found.
- When user clicks the “Generate” button, a spreadsheet document will be generated. When this file is opened in compatible spreadsheet software, user can view the test result in first worksheet (Result). Number of failed and passed tests will be shown, along with individual test case result.

Screenshot



Simple Spreadsheet Test Case 0.1

Formula: =A1+A2

Result On Cell: A3

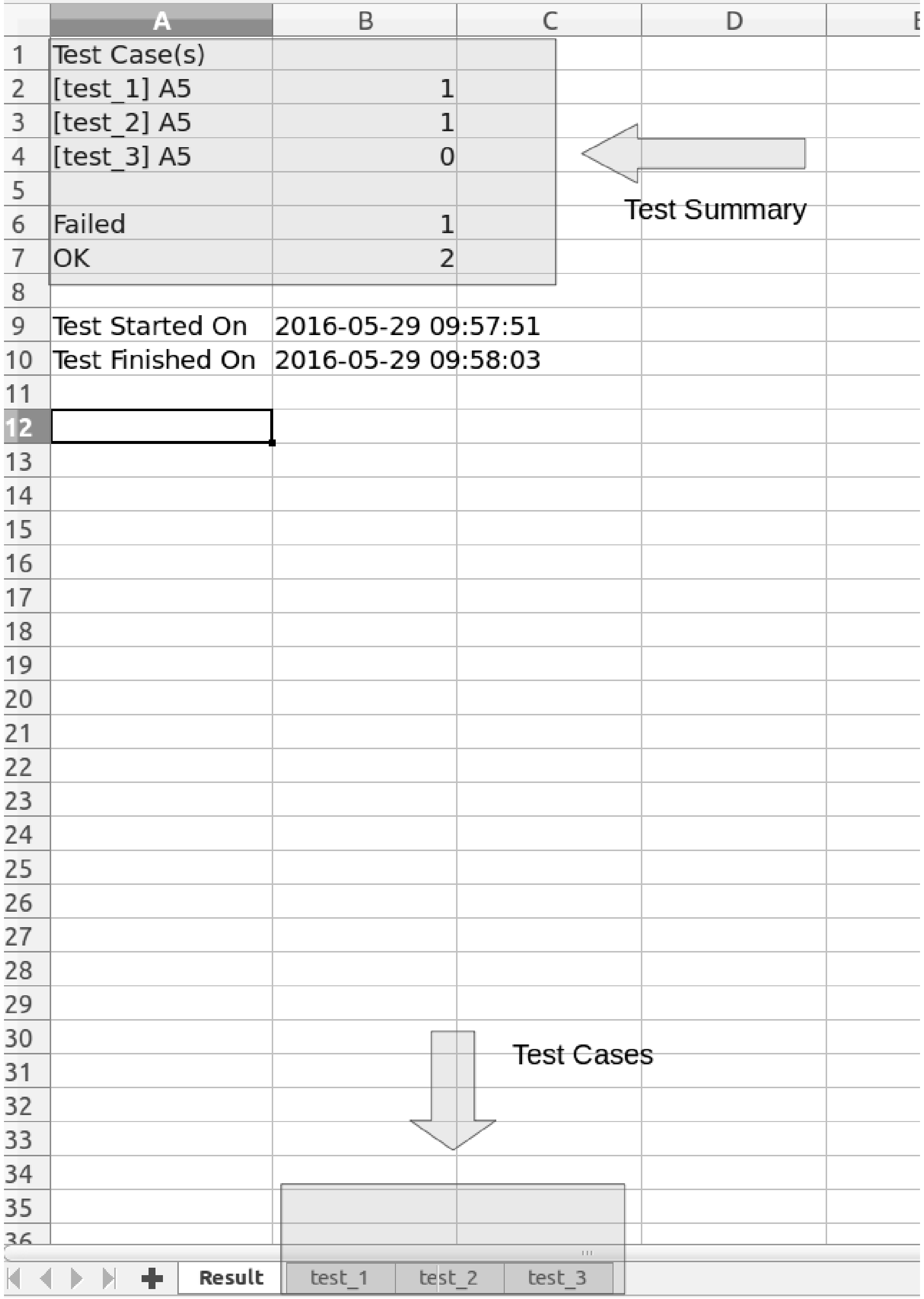
Assert Result On Cell: A5

Compare Formula Result and Expected Result

	Cell Name	Case 1	Case 2	Case 3	Case 4	Case 5
Expected Result	A4	3	7	10		
Cell 1	A1	1	3	5		
Cell 2	A2	2	4	6		
Cell 3						
Cell 4						
Cell 5						
Cell 6						
Cell 7						
Cell 8						
Cell 9						
Cell 10						
Cell 11						
Cell 12						

Cells

Test Cases



	A	B
1	Test Case(s)	
2	[test_1] A5	1
3	[test_2] A5	1
4	[test_3] A5	0
5		
6	Failed	1
7	OK	2
8		
9	Test Started On	2016-05-29 09:57:51
10	Test Finished On	2016-05-29 09:58:03

Test Summary

Test Cases

About the application

- Implemented using Python Programming Language, using Openpyxl library (for working with Office Open XML spreadsheet) and PySide (for user interface).
- Released as free/open source software and may be downloaded from http://noprianto.com

Thank you

References (in the paper)

1) R. Abraham and M. Erwig. GoalDebug: A spreadsheet debugger for end users. In Proceedings of the 29th international conference on Software Engineering (pp. 251-260), IEEE Computer Society, May 2007.

2) R. Abraham and M. Erwig. UCheck: A spreadsheet type checker for end users. Journal of Visual Languages & Computing, 18(1), 71-95, 2007.

3) T. Antoniu, P. A. Steckler, S. Krishnamurthi, E. Neuwirth, and M.Felleisen. Validating the unit correctness of spreadsheet programs. In Software Engineering, 2004, ICSE 2004. Proceedings. 26th International Conference on (pp. 439-448), IEEE, May 2004.

4) M. Burnett. What is end-user software engineering and why does it matter?. In End-User Development (pp. 15-28), Springer Berlin Heidelberg, 2009.

5) M. Burnett, C. Cook, O. Pendse, G. Rothermel, J. Summet, and C. Wallace. End-user software engineering with assertions in the spreadsheet paradigm. In Proceedings of the 25th international conference on Software engineering (pp. 93-103), IEEE Computer Society, May 2003.

6) H. Erdogmus. On the effectiveness of test-first approach to programming, 2005.

7) M. Erwig, R. Abraham, S. Kollmansberger, and I. Cooperstein. Gencel: a program generator for correct spreadsheets. Journal of Functional Programming, 16(03), 293-325, 2006.

8) M. Fisher II, G. Rothermel, D. Brown, M. Cao, C. Cook, and M. Burnett. Integrating automated test generation into the WYSIWYT spreadsheet testing methodology. ACM Transactions on Software Engineering and Methodology (TOSEM), 15(2), 150-194, 2006.

9) A.J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, and B. Myers. The state of the art in end-user software engineering. ACM Computing Surveys (CSUR), 43(3), 21, 2011.

10) H. Lieberman, F. Paternò, M. Klann, and V. Wulf. End-user development: An emerging paradigm. (pp. 1-8), Springer Netherlands, 2006.

11) B.A. Nardi and J.R. Miller. The spreadsheet interface: A basis for end user programming. Hewlett-Packard Laboratories, 1990.

12) R.R. Panko. What we know about spreadsheet errors. Journal of Organizational and End User Computing (JOEUC), 10(2), 15-21, 2005.

13) S.G. Powell, K.R. Baker, and B. Lawson. Errors in operational spreadsheets. Journal of Organizational and End User Computing (JOEUC), 21(3), 24-36, 2009.

14) S.G. Powell, K.R. Baker, and B. Lawson. A critical review of the literature on spreadsheet errors. Decision Support Systems, 46(1), 128-138, 2008.

15) G. Rothermel, L. Li, C. DuPuis, and M. Burnett. What you see is what you test: A methodology for testing form-based visual programs. In Proceedings of the 20th international conference on Software engineering (pp. 198-207). IEEE Computer Society, April 1998.

16) K.J. Rothermel, C.R. Cook, M. Burnett, J. Schonfeld, T.R. Green, and G. Rothermel. WYSIWYT testing in the spreadsheet paradigm: An empirical evaluation. In Software Engineering, 2000. Proceedings of the 2000 International Conference on (pp. 230-239). IEEE, 2000.

17) A. Rust, B. Bishop, and K. McDaid. Investigating the potential of Test-Driven Development for spreadsheet engineering. arXiv preprint arXiv:0801.4802, 2008.