Can Named Ranges Improve the Debugging Performance of Novice Spreadsheet Users?

Ruth McKeever

SToRC
Dundalk Institute of Technology
CatherineRuth.McKeever@dkit.ie

Kevin McDaid

SToRC
Dundalk Institute of Technology
Kevin.McDaid@dkit.ie

Brian Bishop

SToRC
Dundalk Institute of Technology
Brian.Bishop@dkit.ie.

Keywords: POP-I.C. End-User (Office) Applications, POP-II.A. Novices, POP-II.B. Debugging, POP-III.B. Spreadsheets

Abstract

This paper analyses the suitability of certain refactoring methods, in particular those concerning naming, to spreadsheet development, and the effect they can have on both spreadsheet quality and the behaviour of novice end-users. Spreadsheets are normally developed by the end user, and are rarely designed beforehand. They are extremely error prone and often business critical. While naming of ranges in spreadsheets is a recommended best practice, there is little supporting research. This paper describes an experiment to investigate the impact of range names on the debugging performance of novice spreadsheet users.

1. Introduction

This study is part of ongoing research into how refactoring methods, as used in software engineering, can be applied in practice to support the development of better quality spreadsheets.

As financial regulation comes into the spotlight throughout the world, spreadsheet quality is more crucial than ever. Many high-profile errors, such as a sorting error in 2005 that caused aspiring police officers to be incorrectly informed that they had passed an exam (EuSpRIG, 2009), are attributed to a lack of control in spreadsheets. Legislation such as Sarbanes-Oxley, which has been introduced in the US to enforce financial controls on spreadsheets used in business, has made it unacceptable for sizable errors to pass through the auditing net. These errors are partially due to the fact that the majority of spreadsheets are created by their end-user, and rarely by a professional developer. Baker et al (2006) found that only 6% of development time is spent testing spreadsheets, while a study by Powell et al (2007a) discovered errors in 94% of spreadsheets, and 1-2% of cells. Another study by Powell et al (2007b) found that of 25 spreadsheets examined one included errors with an impact of greater than \$100 million. Despite the many attempts made to develop tools and best practices that would reduce the high error rates found in spreadsheets, such as WYSIWYT (What You See Is What You Test) (Rothermel et al, 2000), and U-Check (Abraham & Erwig, 2007), the research suggests that serious errors continue to occur.

Agile programming is based on several values (Agile Manifesto, 2001), two of which are “Working software over comprehensive documentation”, and “Responding to change, over following a plan”. In these two respects spreadsheet engineering is closely aligned to the agile development process. Baker et al (2006) cites Cragg and King (1993) that over 85% of spreadsheets had been modified after their initial implementation and models are updated an average of 7 times. This is the motivation for using agile practices in an attempt to improve the quality of spreadsheet development.

Refactoring is a feature of Agile Software Engineering, which involves cleaning code by making small changes to improve internal structure without changing the external behaviour of the program [Fowler M. 1993]. These small changes increase the understandability of the code and make the errors more obvious. The two refactoring methods looked at in detail for the purposes of this study are

Rename Method and *Replace Magic Number with Symbolic Constant*. *Rename Method* is based on the principle that a method name should reflect its purpose. Developers are encouraged to think about what a comment would say about the method, and rename the method accordingly. Likewise, a formula in Excel can be renamed to reflect its purpose. The *Replace Magic Number with Symbolic Constant* method aims to eliminate the unnecessary hard coding of numbers into software, as this practice frequently leads to bugs. Instead the number should be assigned to a variable, which can then be changed in one place instead of throughout the code. Constants in a spreadsheet can be named in exactly the same way so that they can be changed in one place rather than in every cell that uses them.

The use of Beacons in code is widely believed to increase comprehension for expert programmers. Crosby et al (2002) state that "a plausible definition of a Beacon is a typical indicator of a program's functionality". This is comparable to the role of range names in spreadsheet design. Gellenbeck and Cook (1991) state that "when the procedure body contained a meaningful name, high-level comprehension was achieved much faster." This conclusion forms the basis for the hypothesis that range names can also aid high-level comprehension, and hence understandability, in spreadsheets.

2. Spreadsheet Engineering

This research examines if named ranges can make a spreadsheet easier to understand, and therefore easier to debug. The experiment detailed in this study is an assessment of the performance of novice users debugging a spreadsheet, seeded with errors, that makes extensive use of named ranges (all the formulas in the spreadsheet use names). The authors also question the frequency of use of range names in real-world spreadsheets, and examine a repository of sample spreadsheets in order to answer this.

2.1. Quality

Quality of spreadsheets in this study refers to reliability, understandability, testability and extendibility. The reliability of a spreadsheet is essentially the accuracy of the data that it produces, and is compromised by the errors found in approximately 94% of spreadsheets. Understandability refers to how easily a user or auditor can make sense of the spreadsheet, and is fundamental to the implementation of Sarbanes-Oxley. Testing is crucial if the reliability of the spreadsheet is to be proven, yet is next to impossible if the spreadsheet is not understandable. Extendibility relies on the previous characteristics, yet is vitally important in light of how frequently spreadsheets are reused and remodelled.

2.2. Named Ranges

A range is an individual cell, or group of cells. By naming a range it can then be referred to in formulas throughout the spreadsheet in the same way that a variable is named in software code. By giving a range a meaningful name, as one would give to a variable or method in code, it is believed that formulas will become clearer to the user, therefore more understandable and testable. Without meaningful range names the user must remember the meaning of a cell named, for example "H79" and then check for its occurrence in formulas throughout the workbook. E.g. the formula "netProfit = grossProfit – expenditure", at first glance is far more understandable than "E80 = A40 – D69". A developer can name a single cell, a group of cells, a constant, or a formula. By naming a cell or group of cells you can paste that range elsewhere in the spreadsheet simply by referring to the name. If the original cells are subsequently moved the name will still refer to the same values, hence the pasted values will remain correct. It is possible to perform calculation on a range of cells as an array, using the keys Ctrl+Shift+Enter, however caution must be taken when using a formula on an array of cells, as the resulting array is difficult to change. Array formulas were not used in this experiment.

Constants can be named without needing a cell reference, the developer simply goes to Insert>Name>Define and in the "refers to" box enters the value to which the name should refer. Formulas can be defined in exactly the same way. With formulas the developer has the option of using absolute or relative referencing, but names are absolute by default. Names are commonly used at workbook level, but they can also be declared at sheet level so that one name can refer to the same range of cells on several sheets. Named ranges can also be used for data validation by simply naming a range of allowed values and then using this name as the source for the validation list. To go to a named range in a workbook the user can either click on the name box in the top left-hand corner of the window, or press *F3* on the keyboard, and select which range they wish to go to. This brings the focus directly to the named range. The user can also insert a list of all the named ranges into a worksheet by pressing *F3* and choosing *Paste List*.

Dynamic named ranges allow the developer to name a range where the size of the range is not known from the outset, or if the size may change. To do this the developer defines a name and uses the OFFSET() function to assign a range to the name. This function is used to count how many values are contained in the range, so that if a new row or column is inserted that the calculation is updated.

The inclusion of the Name Manager in Excel 2007 greatly improves the ease with which the user can add, modify and delete names. It also provides the facility for the user to sort and filter the names, and provides a quick insight to what each range refers.

2.3. Research Questions

To establish if Named Ranges are a suitable method for improving spreadsheets, the following research questions were developed:

- Do industrial, academic, standards and training organisations advocate the use of named ranges, and if so, how should they be used and why?
- To what extent and in what way are range names used in practice?
- Does the use of named ranges have an impact on the debugging performance of novice users?

The first two questions are answered in this section while the third question is examined in section 3.

Do industrial, academic, standards and training organisations advocate the use of named ranges, and if so, how should they be used and why?

Both the SSRB (Spreadsheets Standards Review Board) (Hutchens, 2005) and IBM (Read & Batson, 1999) advise the use of named ranges. IBM (Read & Batson, 1999) state that "allocating meaningful range names to areas or cells within a spreadsheet can speed up the development process, make the model easier to understand and reduce the risk of errors made by referring to the wrong cell." They also suggest naming constants rather than hard-coding them into cells, as this makes them easier to change, and recommend referring nearby cells by cell reference and far-away formulas by range name. SSRB, in their Best Practice Spreadsheet Modelling Standards (Hutchens, 2005), describe detailed naming principles, including four naming conventions related to range naming, "Every range name in the workbook should describe the content or use of the range being named". They provide a list of prefixes that should be used with different types of names.

Range naming is recommended by many online spreadsheet courses, (Pearson, 2008), (Mr Excel, 2007), (Ozgrid, 2008) and (Johnson, 2008), and by Microsoft in their online Excel documentation (Microsoft, 2008). As far back as 1985 the Journal of Accountancy ran an article (Bromley, 1985) in which the author states: "The last overall technique is to define names for cell ranges. This reduces the probability of cell reference errors caused by moving a column or row to another location." Bewig (2005) advocates the proper construction of range names for eliminating the problem of referring to the wrong cell while constructing formulas, and states that "well-chosen names are the first and best form of documentation."

An article on Spreadsheet Accuracy Theory (Kruck & Sheetz, 2001) also describes how naming ranges can help the developer comprehend cell meaning "As cells become more highly interconnected, the developer spends inordinate time trying to remember the meaning of the cell, and she/he is distracted from using it effectively. Naming of ranges (Miller 1989) and structure (Ronen Palley Lucas 1989) helps with this problem." The author sums the research by saying "simple policies requiring separation of data areas and user interface areas, requiring cell naming, limits on formula length, or documentation of tests conducted would result in more accurate spreadsheets".

Despite consistent recommendations from the experts to name ranges, there is no academic research at present that examines whether the use of names in spreadsheets increases quality. A survey was conducted of spreadsheet users in Australia (Hall M. 1996), which found that 60% of the surveyed participants used named ranges. The author then stated that 75% should have used named ranges, without any elaboration as to why they should be used. Panko & Ordway (2005) noted that range names are commonly recommended despite a lack of research. The authors warned of the dangers of pointing errors, where a range name was assigned to an incorrect range, and state that "using range names should be considered potentially dangerous until research on using range names is done."

To what extent and in what way are named ranges used in practice?

To establish how named ranges are currently used, a quantitative evaluation of existing spreadsheets was developed. This analysis is currently underway on the EUSES Corpus (Fisher M. 2007), a repository of 5606 real-world sample spreadsheets, including 4498 unique spreadsheets, available to spreadsheet researchers. Initial results show that few spreadsheets use range names in formulas. Currently 1900 spreadsheets from this repository have been analysed, and 27 (1.4%) of these used names in formulas.

3. Experiment

Does the use of named ranges have an impact on the debugging performance of novice users?

In order to answer this research question an experiment was designed to assess how novice users perform debugging a spreadsheet seeded with errors that was developed using range names wherever possible. This experiment was based on an experiment previously carried out by Bishop and McDaid (2007), which was used as the control group data for this study. The reasons for using this control group are outlined below in section 4.3.

3.1. Conducting Experiment

In this experiment a group of novice spreadsheet users examined a spreadsheet that used named ranges and corrected any errors that they found. The performance of this group was then compared with a comparable group who, in related research, performed the same task on a spreadsheet identical in all aspects except that it did not use named ranges. Analysis later in this paper will assess whether the performance of the groups indicates an inherent difference in ability or knowledge.

This original experiment was described in (Bishop & McDaid, 2007) and compared the debugging capabilities of expert versus novice spreadsheet users. It is based on another study conducted by (Howe & Simpkin, 2006) and consisted of a spreadsheet that contained three sheets: *Payroll*, *Office Expenses* and *Projections.*

Payroll calculates the payroll expenses for a typical week, *Office Expenses* sums office expenses for the first quarter and estimates the same for the remainder of the year, and *Projections* estimates the total expenses, both office and payroll, for the next 5 years. It was seeded with 42 errors, and contained no range names.

While the group debugged the spreadsheet, a "time-stamped cell activity tracking tool", known as T-CAT (Bishop B., McDaid K. 2008), ran in the background recording data about each cell click. This tool was developed as a macro in VBA in order to record the interaction between the participant and the spreadsheet with minimal intrusion. This data recorded included each cell entered, at what time the cell was entered, and what changes were made. The resulting data was printed to a hidden worksheet when the spreadsheet was closed and then analysed to examine each participant's interaction with the spreadsheet.

For our study the original structure and format of the spreadsheet was retained. The only change was to name all the ranges in the spreadsheet. Individual cells were named rather than arrays, because of the difficulties faced when attempting to change a value in an array, as previously mentioned. Using formulas on arrays would also have made it impossible for all 42 errors in the original spreadsheet to be replicated in the spreadsheet for our study. This was crucial, as the control group would not have been valid.

This experiment group consisted of 21 second year computing students in Dundalk Institute of Technology, who had a year earlier taken a class in spreadsheet basics, such as formulas and charts. This experiment was carried out in accordance with the ethics policy in Dundalk Institute of Technology. The participants were first given a tutorial on naming ranges, composed of a presentation and a practical exercise that asked them to create, edit, use and delete range names. The authors monitored their performance in this introductory task and were satisfied that they were able to use named ranges to the extent required to perform the experiment. As shall be shown, their performance in the task clearly supports this assumption. When this was completed satisfactorily, they were asked to open up the experimental spreadsheet and begin the trial.

The students chosen for this study were computing students and therefore already understood the concept of variables. Spreadsheet users from a different background are unlikely to understand this principle so easily. The group consisted of a mixture of Irish and international students. There was some concern that the meaning of the range names might not be clear to the students for whom English is not their first language. The initial analysis separated the students but as there was no significant difference in performance between the different nationalities, the following results are based on the groups' performance as a whole. Each student was given an instructions sheet detailing the rules and assumptions that the data in the spreadsheet was expected to follow. They were asked to correct the errors directly on the spreadsheet. The students were not given a time limit for the trial, although it was expected that they would take no longer than an hour.

3.2. Errors

The errors seeded in this trial can be divided into four categories: *Clerical/Non material*, *Rule Violation*, *Data Entry* and *Formula*. There were 4 *Clerical* errors, which consisted of spelling mistakes, and 4 *Rule Violation* errors, which occurred when the data in the spreadsheet violated company policy as detailed in the instructions sheet. There were 8 *Data Entry* errors, for example where a number was entered incorrectly. The 26 *Formula* errors were further divided into four sub-categories: *logic* (9), *cell* (7), *range* (7) and *remote* (3). *Logic* errors consisted of illogical formulas, and instances where a formula was expected but a number used instead. *Cell* reference errors consisted of errors where an incorrect individual cell was referenced while *range* errors consisted of errors where an incorrect group of cells were referenced. *Remote* reference errors occurred where an incorrect reference to a different sheet was used.

4. Results

The results showed that the students corrected on average 47% of the errors seeded in the spreadsheet. This is approximately 11% less than the control group. Unsurprisingly, the only statistically significant difference in performance relates to the debugging of formula errors, where the experiment group corrected only 44% of errors, while the control group corrected 63%.

Error Type	No. Of Errors	% Corrected By Experiment Group	% Corrected by Control Group	Experiment Compared to Control
Clerical	4	11%	11%	0%
Rule Violation	4	63%	65%	-2%
Data Entry	8	64%	63%	1%
Formula	26	44%	63%	-19%

Table 1 – Error Correction Results.

The similar results in the *Clerical*, *Rule Violation* and *Data Entry* categories confirm that the control group and experiment group are comparable in terms of ability and knowledge. As the experimental group performed significantly worse in the formula errors, these were examined further to establish which type of formula errors posed the most problems.

Error Sub-Type	No of Seeded Errors	% Corrected By Experiment Group	% Corrected by Control Group	Experiment Compared to Control
Logic	9	54%	63%	-9%
Cell	7	39%	68%	-29%
Range	7	47%	71%	-24%
Remote	3	19%	28%	-9%

Table 2 – Formula Error Correction Results.

While the experiment group performed worse in all the formula sub-categories, the greatest difference was with *Range* and *Cell* reference errors and these were the only groups where the results were statistically significant at a 5% level based on a non-parametric rank sum test. In each case of these types of error, the control group performed better than the experiment group.

These are the precise category of error that range names were expected to reduce, as these errors are all due to either an incorrect name used in a formula (e.g. the formula in *Payroll F11* is "=GriffinPayRate * HartfordRegularHours" instead of "=GriffinPayRate * GriffinRegularHours"), or a name referring to the wrong cell or range (e.g. the formula in *Office Expenses F18* is "=SUM(FixedYearEst)", but this name refers to an incorrect range of cells).

4.1. Coverage

Coverage maps, based on the coverage tool developed by Bishop & McDaid (2008), were created to visually display what percentage of subjects entered each cell. This tool uses the time values recorded by the T-CAT tool to calculate what percentage of participants examined each cell. To avoid including cells that had only been passed through but not inspected, the minimum time considered was set as 0.3 seconds. This time limit was set according with inspection time analysis carried out by Bishop & McDaid (2008).

A comparison of the coverage maps of the control and experiment groups show how their interaction with the spreadsheet differed. The experiment group became progressively more concerned with the bottom line as the experiment went forward. The first sheet, *Payroll,* showed the greatest difference between the control group and the experiment group. The experiment group looked at more of the data entry cells than the control group, but less of the formula cells, in particular the cells that computed the totals. There was one notable exception to this, the *Overtime Pay* column, which happened to contain three formula logic errors. The second sheet, *Office Expenses,* showed few differences between the control and experiment groups although the experiment group looked at slightly more data entry cells, and slightly less formula cells. The experiment group did seem to look at more formula cells as they went along though, as they viewed the totals for fixed expenses more than the control group. On the final sheet, *Projections,* the experiment group looked at slightly more of the formula cells than the control group. In this sheet they also looked at the data cells, which were all constants, more than the control group.

4.2 Causes

The results above are surprising given the apparent consensus that range names improve the understandability and reliability of spreadsheets. We next investigate three possible explanations:

- High cognitive load.
- Too much confidence in names.
- Did not understand the error, or did not know how to correct it.

High cognitive load

Because the participants were debugging a spreadsheet that they had not personally developed, it was thought that they would have trouble remembering what each name referred to, and might spend additional time flicking back to check this, especially if the range used was located on a different sheet. They would have to complete two checks, one to see if the correct range name was used, and another to see if the name referred to the correct range. This would result in a higher cognitive load and thus may explain the poor performance. However, the students were taught how to click on a name in the name box (or by pressing the F5 key) and it would bring them directly to that range but it is unclear how many of them used that facility.

Too much confidence in names

Because the students were given a tutorial on naming ranges before the experiment commenced, it is possible that they saw the benefits of names and therefore expected them to be correct. This is difficult to prove, but it appeared that on a students' initial reading of a formula, if the first range name appeared to be correct then they did not inspect any further. For example, an error contained in *Payroll F9* occurred when the working hours of one employee were multiplied by the pay rate of a different employee. Only 14% of the experiment group corrected this error, in comparison with 65% of the control group. The erroneous entry was "=EnglebertPayRate*DanielsRegularHours". The correct entry should have been "=EnglebertPayRate*EnglebertRegularHours". 19% of the experiment group, and 65% of the control group corrected a similar error on the same sheet. It seems that when the participants saw that the first name was correct, they presumed that the formula was correct.

The experiment group performed significantly (23%) better in one *Formula Logic* error, where a number had been used instead of a formula. This indicates that the error was immediately obvious because it contained no names, and was not consistent with other cells that performed the same task.

Did not understand the error, or did not know how to correct it

One threat to the validity of this experiment was the possibility that the participant might not fully understand the concept of names. As they were second year computing students they understood the concept of variables. Before the participants began the trial each one watched a presentation and completed a task that taught them how to use range names. This task involved creating new named ranges, creating names to refer to constants, using these names in formulas and editing existing formulas so that they used names instead of cell references. Only when the authors were satisfied that the students could carry out these tasks were they asked to complete the experiment. This ensured that they understood the concept of names.

To investigate whether there was evidence that participants had insufficient knowledge to utilise range names, we examined in detail how each participant handled each range formula error, for which the debugging performance was comparatively low. In some cases it was thought that the subject might have been able to identify the error but unable to repair it. We hypothesise that this would result in numerous unsuccessful attempts at debugging the cells. While up to 7 attempts were made to correct range errors, of those who attempted these errors the majority succeeded in correcting them. In two cases a participant failed after one attempt, and in one case a participant failed after two attempts. This reinforces the authors' belief that the participants had sufficient knowledge about naming in order to correct the errors.

An error in *Office Expenses F18* referred to an incorrectly defined name on a different sheet. 43% of the experiment group, and 74% of the control group corrected this. This cannot be attributed to a lack of knowledge regarding how to fix the error, as of the 10 participants who attempted to fix the error, 9 succeeded.

4.3 Threats to Validity

Using a separate study as the control group causes the most significant threat to the validity of this study. This approach was taken in order to increase the number of participants taking part in the named ranges experiment. The justification for this approach is outlined below under two headings: domain knowledge and consistency of results.

Domain Knowledge

The spreadsheet used in this experiment was designed so that the participants would not need any domain knowledge. However, as the control group were accounting and finance students it could be argued that they could make use of this domain knowledge and be more familiar with the business logic of the spreadsheet. Based on this the authors hypothesised that the experiment group would find less of the logic errors than the control group. This was found not to be the case as the control group corrected only 9% more formula logic errors that the experiment group.

Consistency of Results

The study chosen for the control group was one of a number of similar experiments carried out on students in Dundalk Institute of Technology. One other study is reported here for the purposes of comparison. Group A was a group of 4th year Software Development students who also completed the original experiment (Bishop & McDaid 2008). In this case the group were divided into a control group and an experiment group, with the experiment group making use of a tool that showed them what cells that they had not yet reviewed. The results shown below for Group A are for the control group only.

Error Type	Group A (Control)	Control Group	Experiment Group
Clerical	13%	11%	11%
Rule Violation	66%	65%	63%
Data Entry	66%	63%	64%
Formula	63%	63%	44%

Table 3 – Comparison of Error Correction Results.

As the results clearly display consistency across all the groups of novice spreadsheet users the control group was deemed suitable by the authors.

5. Conclusion

Despite the indications by the initial research that range names would have a positive effect on debugging performance, the results of the experiment suggest that the opposite may be the case. This is not to say that range names should never be used, but that misuse and overuse of names can cause more problems that they solve. Evidence is needed to back up any claims made about techniques promoted as improving spreadsheet quality. As this study shows, practices that work well in one discipline, such as software engineering, do not always have the same effects when applied to different disciplines.

Finally, this work raises the important questions as to what is the appropriate level and use of range names, which have the potential to improve quality and support extendibility and debugging. It may well be the case that different stakeholders may be ideally served by different approaches to naming of ranges.

5.1. Future work

It has already been established that professionals are significantly better than novices at correcting particular types of errors (Bishop & McDaid, 2007). Both this experiment and a qualitative study should be carried out on professional users, as it is unclear whether names would have the same affect on professionals as they do on novices.

In order to establish if the participants' previous knowledge of variable names affected their performance, this experiment should be extended to cover users from other disciplines. This is important, as we already know that developers do not develop the majority of spreadsheets.

This study focuses entirely on users debugging a spreadsheet that has already been developed by someone else. It is more important to reduce errors from the beginning than when the spreadsheet is in use, therefore a different experiment should be designed to examine if developers include less errors in a spreadsheet when they create it using names to begin with.

6. Acknowledgements

This work is supported by the Technological Sector Research measure, funded under the NDP and ERDF.

7. References

Agile Manifesto (2001) [Online] Available: http://agilemanifesto.org/ [March 2009]

Blackwell, A.F. (1999) How to format PPIG submissions. International Journal of PPIG Administrivia, 1(1), 1-3.

Abraham, R. & Erwig, M. (2007) UCheck: A Spreadsheet Type Checker for End Users. Journal of Visual Languages and Computing, Vol.18 (1) Feb. 2007, 71-95.

Baker, K., Fostor-Johnson, L., Lawson, B. & Powell, S. (2006) A Survey of MBA Spreadsheet Users. [Online] Available: http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html [March 2009]

Bewig, P. L. (2005) How Do You Know Your Spreadsheet Is Right? Principles, Techniques and Practices of Spreadsheet Style. [Online] Available: http://www.eusprig.org [March 2009]

Bishop, B. & McDaid, K. (2007) An Empirical Study of End-User Behaviour in Spreadsheet Error Detection & Correction. Proceedings of the European Spreadsheet Risk Interest Group Conference, 2007.

Bishop, B. & McDaid, K. (2008) Speradsheeet End-User Behaviour Analysis. Proceedings of the European Spreadsheet Risk Interest Group Conference, 2008.

Bromley, R. G. (1985) Template Design and Review: how to prevent spreadsheet disasters. Micros in Accountancy, Journal of Accounting, December 1985, 134-145

Butler, R. J. (2000) Is This Spreadsheet a Tax Evader? How H.M. Customs & Excise Test Spreadsheet Applications. Proceedings of the 33rd Hawaii International Conference on System Sciences

Crosby, M., Scholtz, J. & Wiedenbeck, S. (2002) The Roles Beacons Play in Comprehension for Novice and Expert Programmers. 14th Workshop of the Psychology of Programming Interest Group, Brunel University, pp. 58-72

EuSpRIG, Spreadsheet mistakes – news stories [Online] Available: http://www.eusprig.org/stories.htm [Mar. 24, 2009]

Fowler, M. (1993) Refactoring: Improving the Design of Existing Code. New York: Addison-Wesley.

Fisher, M. & Rothermel, G. (2005) The Euses Spreadsheet Corpus: A Shared Resource for Supporting Experimentation with Spreadsheet Dependability Mechanisms. WEUSE05: 1st Workshop on End-User Software Engineering, 2005, pp. 47-51.

Gellenbeck E. M. & Cook, C. R. (1991) An Investigation of Procedure and Variable Names as Beacons during Program Comprehension. Proceedings of the Fourth Workshop on Empirical Programmers, pp. 65-81

Hall, M. J. J (1996) A Risk and Control-Oriented Study of the Practices of Spreadsheet Application Developers. Proceedings of the 29th Annual Hawaii International Conference on System Sciences

Howe, H. & Simpkin, M.G. (2006) Factors Affecting the Ability to Detect Spreadsheet Errors. Decision Sciences Journal of Innovative Education, January 2006, Vol.4(1)

Hutchens, M. (2005) Best Practice Spreadsheet Modelling Standards. [Online] Available: http://www.ssrb.org/best_practice_spreadsheet_modelling_standards_download.html, [Oct. 22, 2008].

Johnson, L. (2008) Naming Ranges, Constants and Cells in Excel: The Whys and Hows. [Online] Available: http://personal-computer-tutor.com/names.htm, [Dec. 8, 2008]

Kruck, S. E., & Sheetz, S. D. (2001) Spreadsheet Accuracy Theory. Journal of Information Systems Vol.12(2) 2001, 93-108

Panko, R. R. & Ordway N. (2005) Sarbanes-Oxley: What About all the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting. Proceedings of the European Spreadsheet Risk Interest Group Conference, 2005.

Powell, S., Baker, K. & Lawson, B. (2007a) Errors in Operational Spreadsheets. [Online] Available: http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html [March 2008]

Powell, S., Baker, K. & Lawson, B. (2007b) Impact of Errors in Operational Spreadsheets. [Online] Available: http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html [March 2009]

Read, N. & Batson, J. (1999, April) Spreadsheet Modelling Best Practice. [Online] Available: http://www.eusprig.org/smbp.pdf, [Oct. 22, 2008].

Pearson, C. H. (2007) Working with Named Ranges in Excel. [Online] Available: http://www.cpearson.com/Excel/named.htm, Sep 6, 2007 [Nov. 25, 2008].

Rothermel, K. J., Cook, C. R., Burnett, M. M., Schonfeld, J. T., Green, R. G. & Rothermel, G., (2000) WYSIWYT Testing in the Spreadsheet Paradigm: An Empirical Evaluation. Proceedings of the 22nd International Conference on Software Engineering, June 2000, 230-239.

Mr Excel (2007, November) Episode 626 – VBA Naming Ranges. [Online] Available: http://www.mrexcel.com/Excel_VBA_Naming_Ranges_training.html [Dec. 8, 2008]

Ozgrid (2008) Lesson 23 – Using Named Ranges in Excel as an Alternative to Cell References. [Online] Available: http://www.ozgrid.com/Excel/free-training/excel-lesson-23-basic.htm, [Nov. 25, 2008]

Microsoft (2008) Define and Use Names in Formulas. [Online] Available: http://office.microsoft.com/en-gb/excel/HA101471201033.aspx?pid=CH100648431033, [Nov. 18, 2008].