Spreadsheet testing in practice

Roy, Sohon; Hermans, Felienne; Van Deursen, Arie

DOI
10.1109/SANER.2017.7884634

Publication date
2017

Document Version
Accepted author manuscript

Published in
Proceedings - 24th International Conference on Software Analysis, Evolution and Reengineering, SANER 2017

Delft University of Technology
Software Engineering Research Group
Technical Report Series

Spreadsheet Testing in Practice

Sohon Roy, Felienne Hermans, Arie van Deursen

Report TUD-SERG-2017-002

TUD-SERG-2017-002

Published, produced and distributed by:

Software Engineering Research Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

ISSN 1872-5392

Software Engineering Research Group Technical Reports:
http://www.se.ewi.tudelft.nl/techreports/

For more information about the Software Engineering Research Group:
http://www.se.ewi.tudelft.nl/

Note: Accepted for publication in the Proceedings of 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER), 2017, IEEE Computer Society.

Spreadsheet Testing in Practice

Sohon Roy, Felienne Hermans, Arie van Deursen
Dept. of Software and Computer Technology
Delft University of Technology
Delft, The Netherlands
{S.Roy-1, F.F.J.Hermans, Arie.vanDeursen}@tudelft.nl

***Abstract*—Despite being popular end-user tools, spreadsheets suffer from the vulnerability of error-proneness. In software engineering, testing has been proposed as a way to address errors. It is important therefore to know whether spreadsheet users also test, or how do they test and to what extent, especially since most spreadsheet users do not have the training, or experience, of software engineering principles. Towards this end, we conduct a two-phase mixed methods study. First, a qualitative phase, in which we interview 12 spreadsheet users, and second, a quantitative phase, in which we conduct an online survey completed by 72 users. The outcome of the interviews, organized into four different categories, consists of an overview of test practices, perceptions of spreadsheet users about testing, a set of preventive measures for avoiding errors, and an overview of maintenance practices for ensuring correctness of spreadsheets over time. The survey adds to the findings by providing quantitative estimates indicating that ensuring correctness is an important concern, and a major fraction of users do test their spreadsheets. However, their techniques are largely manual and lack formalism. Tools and automated supports are rarely used.**

I. Introduction

Spreadsheets are popular end-user development tools [1]. They are used for a myriad of tasks, from project management and financial modeling, to data analysis and scientific calculations. They are used from small businesses to large multinationals, across all domains [2]. Despite being this popular, they suffer from the problem of error proneness [3]. In critical cases, their error proneness becomes a serious problem. For instance, spreadsheet errors have resulted in consequences like financial loss or loss of reputation[1]. To avoid such occurrences, it is important to address the error proneness of spreadsheets and ensure their correctness.

In software engineering, an established way of addressing errors is testing and its related activities. Inspired by this, our ultimate goal is to help spreadsheet users address error proneness, by providing them with better spreadsheet testing techniques and automation support.

Before we can aid spreadsheet users with better testing techniques and tools, we need to know about the existing test practices. Although spreadsheets are similar to traditional software artifacts [4], the majority of spreadsheet users do not possess the knowledge, background, or formal training of software professionals. So how do they manage to ensure correctness of their spreadsheets? To what extent do they perform testing and what methods do they use? Do they use additional tools or aids?

[1]http://www.eusprig.org/horror-stories.htm

To answer these questions, in this paper, we present a study of existing test practices in the community of spreadsheet users. The study reveals what spreadsheet users think and do when it comes to testing spreadsheets.

Following Creswell's *Exploratory Sequential Mixed Methods* approach [5], we design our study with an exploratory qualitative phase, and a follow-up quantitative phase. In the qualitative phase, inspired by *Grounded Theory* research method [6], we collect information by conducting and analyzing 12 semi-structured open-ended interviews (lasting 30-60 minutes) with industrial spreadsheet users. In the quantitative phase, we conduct a structured online survey completed by 72 respondents, with the questions based on the outcomes of the interviews.

The results show that spreadsheet users ensure correctness of their spreadsheets through 1) a set of manual and largely informal test practices, 2) a set of preventive measures, and 3) a set of manual maintenance practices for ensuring correctness over longer periods. The results also reveal a set of perceptions spreadsheet users have regarding importance of testing, impact and quality of testing activities they perform, and usage of test automation and tools.

The contributions of this paper are:

- A qualitative study of spreadsheet testing in practice.
- A quantitative estimation of the extent and popularity of spreadsheet testing related concepts and activities in the spreadsheet user community.

II. Background

We first consider two well-accepted definitions of *Testing* in the field of software engineering as follows.

"Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results."– Hetzel 1983 [7]

"Testing is the process of executing a program or system with the intent of finding errors."– Myers 1979 [8]

In this paper, particularly in the context of spreadsheets, we define *testing* as any activity 1) for determining whether calculations inside a spreadsheet are providing the required results, and 2) aimed at finding errors inside a spreadsheet. Note that spreadsheets without any calculations, for instance those used as databases, are not in scope of this paper.

III. Experimental Design

A. Goal

Our goal in this paper is to understand what spreadsheet users think and do, when it comes to testing or ensuring correctness of their spreadsheets. As such, we seek answers to the following research questions:

RQ1: To what extent do spreadsheet users perform testing or testing related activities?

RQ2: What perceptions spreadsheet users have about testing and ensuring correctness of their spreadsheets?

RQ3: What other methods apart from testing if any, do spreadsheet users follow to address error-proneness of their spreadsheets?

B. Research Method

Following Creswell's *Exploratory Sequential Mixed Methods* approach [5], we design our study with an exploratory qualitative phase, and a follow-up quantitative phase.

The qualitative phase consists of semi-structured interviews with 12 spreadsheet users. The quantitative phase consists of a structured online survey responded by 72 spreadsheet users. The detailed setups, information about participants, and the results are presented in Section IV and Section V.

IV. Qualitative Phase: Interviews

A. Setup

In our exploratory qualitative phase, we conduct and analyze a series of 12 semi-structured interviews with open-ended questions revolving around our three research questions (Section III.A).

We conduct the interviews, lasting 30-60 minutes, over Skype, and record them. When interviewees progressively provide answers similar to earlier ones, a state of *saturation* [6] is reached, and we stop interviewing.

For the analysis of the interviews, we use a technique stemming from the *Grounded Theory* (GT) research method [6]. GT is a research method that originated in the social sciences, but has recently gained popularity in software engineering research [9]. In this method, semi-structured interviews are analyzed through a process called *coding*: association of coherent units of keywords and excerpts collected from the interview recordings, with a *code* representing their key characteristics [6]. The obtained *codes* are grouped together to form abstractions called *concepts*, which again are grouped together to form higher abstractions called *categories*. The *categories* represent the broad outcomes from the interviews.

Inspired by the GT approach, we collect keywords and excerpts from the interview recordings and perform *coding*. We then organize the *codes* into *concepts*, and eventually group the *concepts* together to form *categories*. We present the resulting analysis and coding schema, along with the outcomes in Section IV.C.

TABLE I
Interview Participants

P#	Country	Role	Domain
P1	NL	Financial controller	Finance
P2	UK	Data analyst	Marketing
P3	MY	Business information analyst	Finance
P4	IN	Business process manager	BPO
P5	NL	Financial controller	Finance
P6	US	Data analyst	Manufacturing
P7	KE	Data analyst	Education
P8	IN	IT Project consultant	IT Infrastructure
P9	BR	Accounts officer	Municipal affairs
P10	NL	Chief technology officer	Energy
P11	BE	SW development and delivery manager	HR
P12	DE	Data analyst	Retail

P#: Participant number, NL=Netherlands, MY=Malaysia, IN=India, KE=Kenya, BR=Brazil, BE=Belgium, DE=Germany

B. Participants

Spreadsheet users differ largely in terms of their background, training, expertise, experience, and industrial domains [4]. Our goal is to support all spreadsheet users in general, and therefore, we opted for a mixed group of participants. Thus, our recruitment strategy for the interviews was a combination of 1) directly approaching individuals whom we knew to be expert spreadsheet users, and 2) open invitation for individuals about whom we did not have any prior knowledge. As open invitation, we used announcements— in Twitter, via newsletters, and in MOOCs (Massive Online Open Courses) on spreadsheet related topics conducted by the second author of this paper.

From the resulting set of interested participants, we conducted the interviews till the interviewees increasingly provided similar answers to previous ones and a state of *saturation* [6] was reached. This resulted in the interviewing of 12 participants (identified as P1–P12 in this paper), from 9 countries across the world, fulfilling 8 different professional roles, in 10 different industrial domains, as summarized in Table I. Two of them, P1 and P5, are among those whom we had directly approached, and the rest responded to our open invitation.

C. Results

Following the analysis technique explained in Section IV.A, the coding schema we developed, comprises 4 top-level categories, 9 intermediate concepts, and 1-7 codes per concept, summing up to a total of 30 codes, as depicted in Figure 1. In the following subsections we provide detailed descriptions of the codes, concepts, and categories along with illustrative quotes from the interviewees.

1) Test Practices: This category comprises the various testing practices followed by spreadsheet users. The codes in this category are grouped into the concepts a) *Testing techniques*, and b) *Test related activities* (Figure 1).

a) Testing Techniques: Results indicate that testing of spreadsheets is not uncommon, as 10 out of 12 (P1, P3-P6,

Codes	Concepts	Categories
C1.Ad hoc manual testing	Testing techniques	Test practices
C2.Testing through comparison		
C3.Simulation		
C4.Random testing		
C5.User testing		
C6.Invariant-based testing		
C7.Refactoring for testing	Test related activities	
C8.Communication of test results		
C9.Importance of correctness	Importance of correctness	Perceptions
C10.Reflection on time and effort spent on ensuring correctness		
C11.Rationales for not testing	Views on testing	
C12.Impact of testing		
C13.Quality of testing		
C14.Nonuse of automation or tools	Views on test automation and tools	
C15.Reflection on test automation and tools		
C16.Adherence to standards and best practices	Development techniques	Preventive measures
C17.Design patterns for quality		
C18.Measure of complexity		
C19.Iterative development		
C20.Language feature selection for quality		
C21.Structured programming		
C22.Data validation		
C23.Manual inspection	Review processes	
C24.Peer review		
C25.Code review		
C26.Access control through locking	Acess control	
C27.Importance of maintainability	Maintenance	Maintenance practices
C28.Documentation		
C29.Manual version control		
C30.Manual changelogs		

Fig. 1. Coding schema with 30 codes, 9 concepts, and 4 categories

P8-P12) interviewees spoke about doing some form of testing on their spreadsheets. As we illustrate below, the approaches in most cases are not as formal as in traditional software testing, and most of the users do not perceive or name the activities as 'testing'. Both of these could be possibly due to the fact that they do not have formal training or foundations in software engineering principles. We present a closer view of the findings as follows.

6 out of 12 interviewees (P3, P5, P8-P10, P12) stated how they manually test correctness of their formulas or VBA code without following any fixed strategy or formalism, which we interpret as the code *C1.Ad Hoc Manual Testing* (Figure 1). Among the various features Excel offers to its users, use of VBA (Visual Basic for Applications) based macros is one. This enables users to write VBA code in order to automate various repetitive tasks in spreadsheets, or perform complex calculations. P5 explains how he tests his VBA code in split screen view, by stepping through the code on the left side and observing changes in the spreadsheet on the right side. P8 talks about how he selects a few formulas from a set of copied across formulas, and manually checks their calculation. P12 even uses a table calculator to check the results of formulas. Apart from such ad hoc manual approaches, we also observe instances of specific strategies being applied during manual testing as follows.

A practice, usually not common in software testing, appears to be popular in the spreadsheet community: comparing outputs calculated via different methods for the purpose of testing, which we interpret as *C2.Testing through Comparison*. 6 out of 12 interviewees (P3, P6-P10) stated of following this strategy, like P8 says *"In some cases there are two ways to getting a solution, write both type of formulas and check if same results are obtained."* According to the interviewees, such comparison can be done 1) between two or more different formulas, 2) between formulas and the summary provided by Excel, or 3) between entirely different spreadsheets.

C3.Simulation is a variation of manual testing using dummy operational scenario data for testing spreadsheets mentioned only by P3.

C4.Random Testing is another manual testing strategy with randomly selected inputs used to test spreadsheets, also mentioned only by P3.

One other variation of manual testing we found, we interpret as *C5.User Testing*, where two interviewees (P3, P5) stated of creating spreadsheets for other users, and then relying on those users to find errors or problems. As P5 says, *"The final testing is done by the users; if they observe abnormalities, they report to me."*

Lastly, we found *C6.Invariant-based Testing* as a common practice among spreadsheet users, as 7 out of 12 interviewees (P1, P3-P5, P8-P10) mentioned of frequently doing so. This form of semi-automatic testing involves the use of a separate set of formulas testing the outputs of an original set of formulas, which are implementing the core functionalities of a spreadsheet. For example, a conditional function like IF testing the outputs of other formulas that are calculating credit and debit, checking for invariant properties like "*Sum of total credit and total debit should always be zero*" or output should never exceed a certain maximum value.

b) Test Related Activities: Here we describe the two codes we found which were not testing techniques *per se*, but are closely related to testing, as follows.

P8 told us about how he sometimes tests his longer calculation chains by breaking them down into smaller pieces, and testing those pieces individually, which we interpret as *C7.Refactoring for Testing*. He explains, *"For instance a bigger calculation may be using 10 different simpler calculations done by formulas inbetween, that calculation can be checked by testing each of the component calculations separately."*

Five of the interviewees (P3-P5, P8, P11) mentioned how they use special formatting, highlighting, or colors to indicate errors detected by testing techniques they are using. Thus, indication of test failures through visual clues appear to be a practice followed by some spreadsheet users, which we interpret as *C8.Communication of Test Results*.

2) Perceptions: The second category emerging from the interviews (Figure 1) sheds light upon how spreadsheet users

think when it comes to ensuring correctness of their spreadsheets. This category consists of the concepts a) Importance of Correctness, b) Views on Testing, and c) Views on Test Automation and Tools, illustrated as follows.

a) Importance of Correctness: For majority of spreadsheet users the *C9.Importance of Correctness* is high. From the interviews we find that, spreadsheets are not only used for data analysis and calculations, but also for directly reporting the results for important decision-making. Thus, wrong results can easily lead to wrong decisions which makes ensuring correctness an important concern. 10 out of 12 interviewees (P1, P3-P11) confirm this as a concern. For instance, P8 who reports to high level government officials like the secretary of a ministry, explains: *"If mistakes emerge in front of the secretary you have to do the whole thing again; that should not happen."*

Our participants believe that they tend to spend considerable amounts of time and effort ensuring correctness of their spreadsheets, as illustrated by the code *C10. Reflection on Time and Effort Spent on Ensuring Correctness*. Five interviewees (P4, P5, P7-P9) explicitly referred to this. For instance as P4 exclaims: *"Lots of time is taken up for checking and ensuring integrity of results, it is tedious."* Only P2 stated on the contrary saying she spends little time or effort, reasoning that the results she produces are only used for the purpose of obtaining overviews or observing trends, and not critical decision-making.

b) Views on Testing: While users understand the utility of testing their spreadsheets, and most of our participants reported of testing (10 out of 12, P1, P3-P6, P8-P12), many of them also put forward reasons for not performing testing at all, or not increasing the extent of their testing, which we interpret as *C11.Rationales for not Testing*. This points to an interesting direction of future work in investigating what are the barriers of spreadsheet testing. This is however, not in scope of this paper as here we focus on how spreadsheet testing is done.

We obtained insights on what users think about the *C12.Impact of Testing* on their work. P2 was curious in this case as she exclaims, *"I have no idea how you can test a spreadsheet! I am curious to know."* On the other hand, P8 was positive stating *"Errors are generally rare because testing is done"* expressing his confidence on testing as a means to reduce errors.

When asked about how satisfied they were with their testing efforts, P1, P10, and P11 stated that they were satisfied, whereas P5 said he just trusts his measures are adequate, and P3 said that he acknowledges the risk but leaves it at that, referring to the general unpredictability of working with spreadsheets– *"There are a lot of possibilities that can happen in a spreadsheet!"* We interpret these views as perceptions on *C13.Quality of Testing*.

c) Views on Test Automation and Tools: Despite the existence of a number of research initiatives in the past [10], [11] our participants remain oblivious to the fact that there are or can exist automated support or tools for testing their spreadsheets. All the interviewees denied using any tool for testing, or the knowledge that such tools exist (*C14.Nonuse of Automation or Tools*). This emphasizes the need for developing tools or support that users can actually use, and not just implement a technique from traditional software engineering in the context of spreadsheets. This is one of our key motivation behind this field study: to ascertain what is actually happening in practice before proposing a method or technique.

Although automation or testing tools are not used by our participants, they do foresee the benefits and generally maintain a welcoming stance for possible innovations. We interpret these views as the code *C15.Reflection on Test Automation and Tools*. Five of the interviewees (P4-P8) expressed their belief that automation for testing will largely help them, as P4 mentions *"It will be very efficient because we will save a lot of time."* P2 was curious how automation is possible. P3 and P9 were unsure how tools may turn out, as P3 states *"I have never thought about that, because a spreadsheet is spreadsheet, not an application."* P3 also mentioned his concerns over increase in cost the tools may cause. Overall, it shows that awareness can overcome whatever minor reluctance there is about using testing tools or automation, and the community is otherwise welcoming to innovation in this regard. This re-inforces our motivation for developing better testing support and tools.

3) Preventive Measures: The third category to emerge from the interviews relates to preventive measures that a minority of spreadsheet users take in order to ensure correctness of their spreadsheets, as a complementary strategy to testing.

Firstly, they follow the concept a) Development Techniques to reduce the chances of errors. Secondly, they follow the concept b) Review Processes, and finally, they make use of the concept c) Access Control, to prevent errors due to unwanted or accidental modifications of spreadsheets. We illustrate these concepts with the findings from the interviews as follows.

a) Development Techniques: Three of the interviewees (P1, P3, P11) mentioned about following general best practices and standards when developing their spreadsheets, as they believe it helps prevent occurrence of errors (*C16.Adherence to Best Practices and Standards*). For instance as P1 states, *"I think it is important as well is how I use spreadsheets. For example, I will never add hard numbers in the formula fields, because the most of the things that go wrong is because you used some kind of correction in your formula, so I dont do that anymore"*, or like what P11 refers to when saying, *"Some best practices, like do not repeat yourself, do not use numbers in formulas etc., common sense things."*

Three interviewees (P3, P5, P11) mentioned using specific design patterns (*C17.Design Patterns for Quality*) to reduce chances of errors, like P3 states, *"Our design has three main sections across three separate worksheets— input, settings, and output. Input for entering data, settings for all formulas, and output for the results."*

We found one instance where the interviewee mentioned of controlling complexity of his formulas (*C18.Measure of Complexity*), as P3 talks about avoiding complex formulas, writing simple formulas, and splitting up complex operations into smaller and simpler ones.

We found an instance of *C19.Iterative Development* as P1 mentions *"I think in the way I try to work that I want to eliminate as soon as possible that there are mistakes in my spreadsheets, checking in every step, what is the data used, what is the output, whether output is expected."*

P3 states how he avoids use of VBA in his spreadsheets to reduce chances of errors, referring to the increase in complexity caused by use of VBA. On the contrary, P5 mentions of using macros (VBA) to avoid errors citing the fact that errors need to be only located in the macro code as rest of the formulas in the spreadsheet are generated automatically, leaving no chance for human errors. Both appears to practice what we interpret as *C20.Language Feature Selection for Quality*

P5 refers to following a *C21.Structured Programming* approach when he says *"I try to make blocks of code [in VBA]."*

Lastly, four interviewees (P5, P7-P9) mentioned about using checks to validate input data in their spreadsheets which we interpret with the code *C22.Data Validation*. As P7 mentions of doing, *"...data cleanup as in verifying data is correct both in terms of format and in terms of content, like missing information, wrongly entered e-mail etc."*, or as P8 states *"Best to check data before starting calculations, data cleansing is an important initial requirement."*

b) Review Processes: 6 out of 12 interviewees (P1, P2, P4, P6, P8, P10) mentioned performing *C23.Manual Inspection* of their spreadsheets to detect problems. The process revolves around visually checking spreadsheets for errors or inconsistencies, as P4 points out *"We check whether all the correct filters are applied, all the regions are in order, locations are in order, nothing extra is showing, nothing important is missing."* Two of the interviewees talk about relying on their intuition and experience to be able to effectively perform this: P10 mentions *"I expect results to lie within certain ranges based on experience"*, and P2 says, *"Gradually with having a better sense, you can spot problems with numbers."* This code is different from *C1.Ad Hoc Manual Testing*, as *C1* involves checking output by feeding various input values, and this code involves just visual inspection of spreadsheets.

C24.Peer Review appears to be another practice spreadsheet users resort to, as 4 interviewees (P2, P3, P7, P8) talk about relying on their colleagues and supervisors for detecting issues in their spreadsheets. For instance, P8 explains, *"Sometimes showing your results to colleagues helps, sometimes it is difficult to spot your own mistakes, so sometimes colleagues can spot problems, peer review or whatever you call it"*, P3 states, *"At least two persons must know how it [spreadsheet] works"*, and P7 mentions, *"My direct supervisor does a cross-check."*

Only one interviewee (P5) specifically refers to *C25.Code Review*, when he talks about performing a *walkthrough* through his VBA code.

c) Access control: Three interviewees (P3, P7, P11) discussed access control through locking mechanisms as a measure to prevent accidental or erroneous modifications to their spreadsheets (*C26.Access Control through Locking*). All three use the *password protection* feature offered by Excel through which spreadsheet files can be locked in workbook or worksheet level.

4) Maintenance Practices: The fourth and final category emerging from the interviews comprises maintenance practices that a small minority of spreadsheet users appear to be following for ensuring correctness of their spreadsheets over longer periods of time. As such, there are no separate concepts for this category. We observe concerns over maintainability of spreadsheets, the practice of documentation, and lastly, manual implementation of version control and changelogs. We illustrate the findings as follows.

C27.Importance of Maintainability: Interviewee P3 explicitly expressed his concerns over maintainability of spreadsheets saying *"...we are really concerned about the sustainability of the [spreadsheet] templates. Most of the templates I create, I have given the target like at least the template should survive more than three years."*

C28.Documentation: Three interviewees (P3, P5, P11) mention using documentation to explain different aspects of their spreadsheets. The documentation can be in the form of 1) separate worksheets dedicated for the only purpose of documentation, or 2) comments inside the worksheets, or in-between lines of VBA code. As P3 says, *"We use a ReadMe worksheet where we explain what are the formulas that have been used in this workbook, who designed it, when we designed it, who and when it was checked [inspected or tested], and when it was last checked"*, or like P5 says *"...and I always try also to document my macros [VBA] very well with those– how do you say it, these lines of information in between, what I want to do, what I intend to do? Yes the comments, exactly!"*

C29.Manual Version Control: Only P3 mentions about manually implementing a version control system by recording each version of spreadsheets in the accompanying documentation worksheets explained above.

C30.Manual Changelog: P3 also mentions manually maintaining a changelog— *"Changelog; we will document the changelog, like for example which cells have been changed."*

V. Quantitative Phase: Survey

Having obtained the set of four categories— Test Practices, Perceptions, Preventive Measures, and Maintenance Practices, as described in Section IV, we have an understanding of what spreadsheet users think and do, when it comes to ensuring correctness of their spreadsheets. In this section, we obtain estimates of the extent to which these practices and perceptions have penetrated the spreadsheet user community.

A. Setup

In the quantitative phase, we conduct a structured online survey, completed by 72 spreadsheet users. The survey consists of 45 questions of which 30 are closed-ended questions related to spreadsheet testing, based on the 30 codes that emerged from the qualitative phase (Figure 1). Of the remaining questions, 9 are about the respondents, and 6 are about debugging, type

TABLE II
MOST FREQUENTLY OCCURRING OCCUPATIONS OF SURVEY RESPONDENTS

Occupation	No. of Respondents
Manager	9
Analyst	9
Researcher	7
Professor	6
Software Engineer	5
Consultant	5
Data analyst	5
Enigineer	5
Auditor	3
Student	3

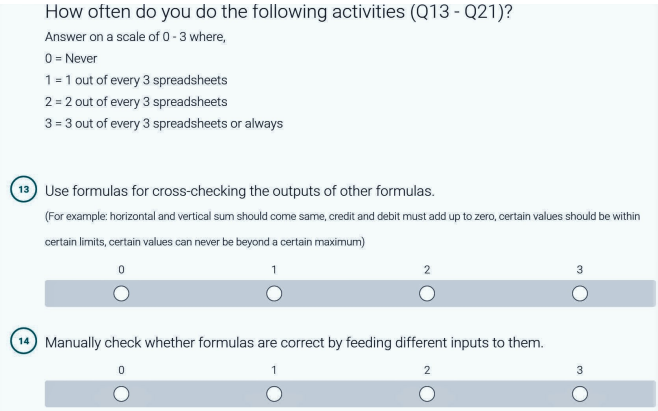

Fig. 2. Example of survey questions depicting 0-3 scale for frequency of usage

of errors, and other topics which are not under scope of this paper. We provide a link to the survey[2].

B. Participants

We announced our survey via Twitter, the mailing list of the EuSpRIG (European Spreadsheet Risks Interest Group), the mailing lists of MOOCs on spreadsheet topics conducted by the second author, and LinkedIn. The respondents who completed the survey were from 21 different countries, of average age 40, with 91% male, and 9% female. The 10 most frequently mentioned occupations of the respondents are shown in Table II.

C. Results

In the following subsections we describe the findings from the survey, organized in the same way as Section IV: one sub-section for each of the categories— Test Practices, Perceptions, Preventive Measures, and Maintenance Practices.

1) Test Practices: In the survey (Q13-Q21), we ask the participants to state how frequently they use the different spreadsheet test practices identified in Section IV, on a scale of 0–3 as depicted in Figure 2. For ease of interpretation, henceforth in this paper, we refer to the steps of this scale as 'Never', 'Sometimes' (1/3), 'Often' (2/3), and 'Always' (3/3).

[2]www.surveymonkey.com/r/spgtres

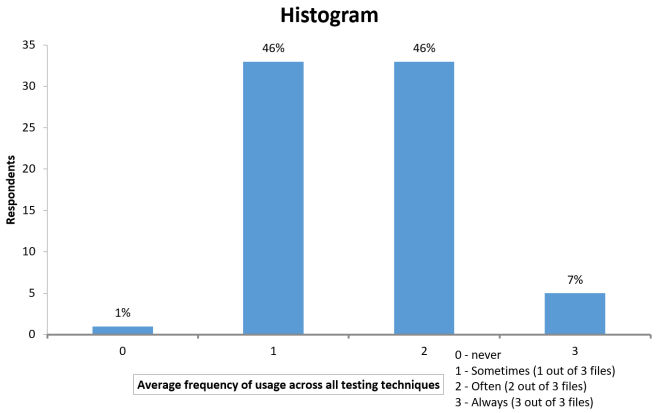

Fig. 3. Number of participants vs. Average frequency of usage across all testing techniques: 46% and 46% of the participants perform some form of testing, with average frequencies in the range of sometimes, and often respectively

TABLE III
TESTING TECHNIQUES

Testing Techniques	Never	Sometimes	Often	Always
C1.Ad hoc manual testing	13%	35%	31%	22%
C6.Invariant-based testing	24%	29%	33%	14%
C2.1.Compare formulas	25%	31%	38%	7%
C2.2.Compare formulas and Excel summary	28%	35%	21%	17%
C3.Simulation	35%	35%	19%	11%
C4.Random testing	39%	33%	17%	11%
C5.User testing	47%	33%	15%	4%
C2.3.Compare spreadsheets	51%	32%	13%	4%

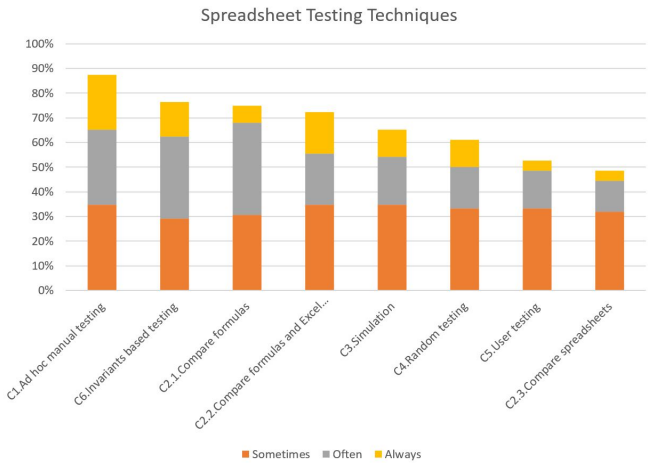

Fig. 4. Percentage of respondents vs. Frequency of usage, for each type of Testing Technique, sorted left to right in descending order of popularity

a) Testing Techniques: First, to obtain a measure of how common testing is in the community of spreadsheet users, we calculate the average frequency of usage for each respondent across all types of testing techniques, using the 0-3 scale values. From the histogram in Figure 3 we see that two equal sets of respondents, 46% and 46%, conduct in average some form of testing, sometimes, and often respectively. This supports our finding from the interviews that testing is common among spreadsheet users. However, only 7% of the respondents perform in average some form of testing always, indicating that only a minority of spreadsheet users treat testing as a mandatory and regular activity. 1% of the respondents stated of not performing any type of testing at all. They are represented by respondents like the one who denied performing any type of testing, citing the reason *"I use spreadsheets mostly for text. My spreadsheet that do have a statistical purpose have a very limited range (i.e. no more than 100-200 numerical entries) and wrong results will not have serious consequences."*

Next, for each type of testing technique (codes C1-C6), the percentage of respondents and their respective choices according to the 0-3 frequency of usage scale is shown in Table III.

From Figure 4, we observe that:

- The most popular testing technique is *C1.Ad Hoc Manual Testing*, which is used by 88% of respondents, and by 22% always.
- The four other popular techniques appear to be *C6.Invariant-based Testing, C2.1.Comparing Formulas, C2.2.Comparing Formulas with Excel Summary*, and *C3.Simulation*. These results also corroborate the findings from the interviews.
- Techniques of *C4.Random Testing*, and *C5.User Testing*, do not seem to have penetrated the community to a large extent, again corroborating the interview findings.
- The least popular practice is *C2.3.Comparing (entirely different) Spreadsheets* for testing, which 51% of respondents state of never following.

b) Test Related Activities: We find that *C7.Refactoring for Testing* is a popular practice with 89% of respondents following it, and 25% following it always.

With more than 86% respondents following, and 22% doing it always, *C8.Communication of Test Results* also appears to be a common practice in the spreadsheet users community.

2) Perceptions: Survey questions Q10-Q12, and Q22-Q26, provide us with insights about the perceptions of spreadsheet users regarding importance of correctness and testing.

a) Importance of Correctness: To ascertain the *C9.Importance of Correctness*, we ask the users about how frequently they work with critical spreadsheets, where errors can have serious consequences. We use the same frequency scale of 0–3 as shown in Figure 2. We find that 87% of respondents work with critical spreadsheets, and 21% do that always. Thus, as noted during the interviews, ensuring correctness of spreadsheets is an important concern for spreadsheet users.

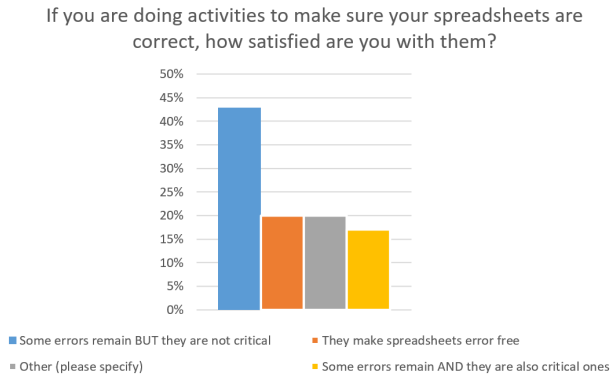

Fig. 5. Percentage of respondents vs. Satisfaction with quality of testing

We investigate how much time the users spend on addressing this concern (C10). We find 29% of respondents spending more than 30%, and 39% spending 20-30%, of their time in ensuring correctness of their spreadsheets or testing.

b) Views on Testing:

C11.Rationales for not Testing: As explained in Section IV.C.2.b, investigation of this code is not in scope of this paper.

C12.Impact of Testing: We find that 90% of the respondents believe that performing test related activities reduces the chance of errors in their spreadsheets.

C13.Quality of Testing: From Figure 5, we observe that 43% of the respondents state that non-critical errors still remain in their spreadsheets, even after whatever testing practice they follow. A further 17% of the respondents paint a graver picture by stating that even critical errors remain in their spreadsheets after they perform testing. In the interviews, only 3 out of 12 participants mentioned that they were satisfied with their testing activities. These results imply that although testing is common among the spreadsheet users, the quality of their tests is not satisfactory leaving ample scope of improvement.

c) Views on Test Automation and Tools: Regarding usage of automation or tools (C14) for testing, 69% of respondents answered negatively, and 31% answered positively, re-iterating the fact that major portion of testing in the spreadsheet community is still done manually providing us the motivating for developing better tools and supports.

From Figure 6, we observe that 73% of the respondents believe automation and tools for testing can help reduce errors (C15). We also note that 34% are apprehensive about increasing cost due to tools, and 29% suspect tools may make their work complicated.

3) Preventive Measures: Questions Q27-Q36 of the survey provide us with estimates of how popular the different preventive measures identified in Section IV.C.3 are.

a) Development Techniques: We ask the respondents how frequently they use the development techniques identified in Section IV.C.3, according to the scale of 0-3 as shown in Figure 2. The percentage of respondents and their respective choices for each technique is shown in Table IV.

From Figure 7, we observe that

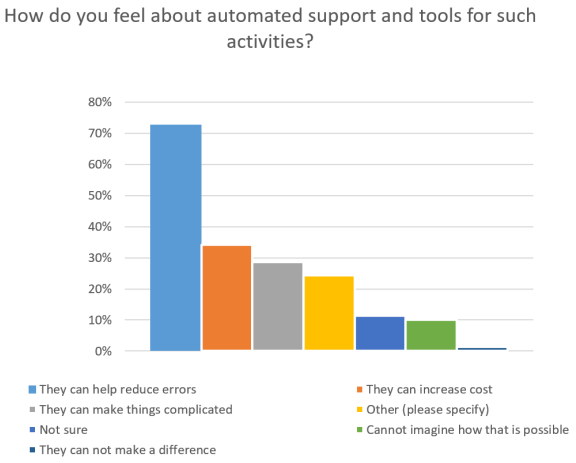

Fig. 6. Percentage of respondents vs. Opinion about automation and tool support for testing

TABLE IV
DEVELOPMENT TECHNIQUES

Development Techniques	Never	Sometimes	Often	Always
C19.Iterative development	3%	25%	32%	39%
C18.Measure of complexity	10%	29%	36%	26%
C20.Language feature selection for quality	19%	16%	22%	43%
C16.Best practices, standards and C17.Design patterns for quality	21%	23%	30%	26%
C22.Data validation	30%	27%	25%	18%

- The most popular technique is *C19.Iterative Development* with 97% of respondents using it, within whom 39% use it always.
- All of the techniques are fairly common, since the least popular technique of *Data Validation* is also practiced by 70% of the respondents, of whom around 18% do it always.

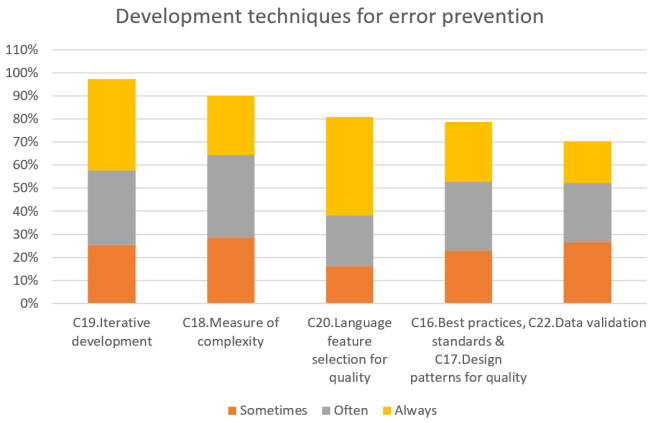

Fig. 7. Percentage of respondents vs. Development techniques, sorted from Left to Right in descending order of popularity

Among the respondents who use VBA (n=37), 81% practice structured programming, of whom 49% practice it always.

b) Review Processes: Both *C23.Manual Inspection* of spreadsheets and *C24.Peer Review* appear as common practices. 97% of respondents perform manual inspection, within whom 48% perform it always. In contrast, 68% practice peer review, within whom only 6% practice it alway. Thus, manual inspection is more common between the two.

Among respondents who use VBA (n=37), 83% perform *C25.Code Review*, within whom 67% perform it always.

c) Access Control: From the interviews we learned about participants following access control to prevent errors in their spreadsheets in the form of locking (C26). From the survey we find that the practice is common, with 60% indicating of locking through password protection.

4) Maintenance Practices: From Q37-Q41 of the survey, we estimate the popularity of the maintenance practices identified in Section IV.C.4.

To obtain a measure of how important maintainability of spreadsheets (C27) is, we ask the respondents how frequently they work with spreadsheets that remain in use for more than 6 months, and 12 months, according to the 0-3 frequency scale as shown in Figure 2. We find that spreadsheet maintenance is important: only 5-10% respondents have never worked with spreadsheets that are in use for over 6 months. It is more common to work with spreadsheets in use for over 6 months, which 70% of the respondents confirm as doing either sometimes or often. It is less common to work with spreadsheets in use for over 12 months, for which the same population is reduced to just below 50%.

Next, we ascertain which maintenance practices identified during the interviews are more popular. We find that *C28.Documentation* and *C29.Manual Version Control* apparently are prevalent practices with followers in the range of 80% and 70% respectively. *C30.Manual Changelog* however, is not that popular, as below 40% of respondents indicate of practicing it.

VI. REVISITING THE RESEARCH QUESTIONS

In this section, we revisit our research questions in the light of results obtained.

RQ1 ***To what extent do spreadsheet users perform testing or testing related activities?***

From the category Test Practices, we learn that testing spreadsheets is common: 10 out of 12 interviewees and 92% of 72 respondents in the survey do some form of testing. However, there are only six techniques in use, of which ad hoc manual testing is the most popular, followed by invariant-based testing, and testing through comparison of different formulas. Usage of simulation, random testing, and user testing is relatively low. Apart from testing techniques, refactoring for testing and communicating test results through special formatting, highlighting, and colors is also common practice, indicated by 89% and 86% of the survey respondents respectively.

RQ2 ***What perceptions spreadsheet users have about testing and ensuring correctness of their spreadsheets?***

From the category Perceptions, we learn that:

1) Ensuring correctness is an important concern for spreadsheet users. 87% of respondents indicate working with critical spreadsheets, within whom 21% are doing it always.
2) Users also spend considerable time in ensuring correctness with 29% of respondents spending more than 30% of their time with spreadsheets on this.
3) 90% of respondents believe that performing testing activities reduces chances of errors.
4) Quality of testing activities can be improved as 43% of respondents state that errors remain even after performing testing, and a further 17% state that even critical errors remain.
5) Usage of tools and automation for testing is uncommon as 69% of respondents indicate never having used any tool. This calls for developing better tools and support for spreadsheet testing.
6) Although tools are presently uncommon, 73% of the respondents believe they would prove helpful, while 34% and 29% are apprehensive about increase in cost and complication of work respectively.

RQ3 ***What other methods apart from testing, do spreadsheet users follow to address error-proneness of their spreadsheets?***

From the two categories, Preventive Measures and Maintenance Practices, we learn what spreadsheet users do apart from testing to address errors.

As measures to prevent errors, a set of development practices are followed among which the iterative development is most popular. Reviewing practices are also common with 97% of respondents indicating that they conduct general manual inspection of their spreadsheets. From the interviews, we learn that manual inspection benefits from users' growing intuition with increase in experience. Peer review is also common indicated by 68% of survey respondents. Lastly, we observe use of access control with 60% indicating usage of locking mechanism like password protection of specific worksheets in a spreadsheet.

We learn that maintenance of spreadsheets is important. 70% and 50% of respondents indicate of working with spreadsheets that remain in use for over 6 months, and over 12 months respectively. In order to ensure correctness of spreadsheets over longer periods, users follow a set of maintenance practices of which documentation and manually implementing version control is common, practiced by respondents in the range of 80% and 70% respectively. Manually maintaining changelogs is not so common with 40% indicating of practicing it.

VII. Related Work

The topic of spreadsheet testing has been explored in the past. Jannach *et al.* present an overview of various approaches for spreadsheet QA including testing approaches [12]. Notable is pioneering work by Rothermel *et al.* who proposed the WYSIWYT approach [11] for spreadsheet testing. In this paradigm, spreadsheet users have to mark formula outcomes as correct or incorrect, after which the WYSIWYT system calculates which formulas led to the checked values and increases their testedness. This paradigm was subsequently enhanced through the work of Fisher *et al.* who proposed an approach for automated test case generation of spreadsheets [13], and later integrated that approach into the WYSIWYT framework [14]. Another approach for automated test case generation was implemented by Abraham *et al.* in [15] demonstrating improvement compared to [13] using the same experimental setup. Related is also the work of Burnett on spreadsheet assertions that allows spreadsheet users to define assertions and propagates them through cell dependencies [16]. Other studies have confirmed the applicability of testing to spreadsheets [17]. More recently, McDaid *et al.* explored the possibility of applying test driven development in the context of spreadsheets [18].

Closely related is work by Panko [19] in which the author recommends spreadsheet test practices. Similar is work by Pryor [20] in which the author discusses his views and experiences related to spreadsheet testing. These works however, do not investigate what practices are being followed by spreadsheet users at large.

Most closely related is the work by Hermans [21]. In this work, the author provides insights into existing spreadsheet test practices through analysis of spreadsheet corpora.

None of these works however, attempt to systematically investigate existing test practices in the spreadsheet user community, which we describe in this paper. In the course of our study, we also did not find any evidence of the previously proposed testing approaches being used by the spreadsheet users, emphasizing the need for tools and techniques users can actually adopt and use. In this regard, we believe, an understanding of the present practices would prove vital, which is one of the key motivations behind the study presented in this paper.

VIII. Threats to Validity

A. Threats to External Validity

A threat to external validity of our results concern the representativeness of the participants. This threat applies in particular to the interview participants. We have therefore used a mixed group of interviewees recruited both through direct approach and open invitation. The interviewees are also spread across 9 countries and from 8 different professional roles. Most importantly, we conducted the survey completed by 72 respondents, to obtain affirmation of the interview outcomes and mitigate this threat.

B. Threats to Internal Validity

A threat to internal validity of our results arises in relation to our conducting of the interviews and the analysis we performed based on our interpretations. We tried to minimize biasing the interview participants by keeping our questions

open-ended and providing them ample freedom of expression. As far as the analysis is concerned, the process of coding the interviews, and the categorization thereof is subject to our own interpretations, but we tried to attain as much commonality as possible through repeated discussion and brainstorming between the three authors.

IX. Concluding Remarks

The objective of this work is a systematic study of spreadsheet testing in practice. In this paper, we have described a mixed methods experimental setup with an exploratory qualitative phase and a follow-up quantitative phase. In the qualitative phase we interviewed 12 industrial spreadsheet users. The outcome of the interviews, organized into four categories consists of test practices, perceptions about testing and correctness, preventive measures, and maintenance practices. Based on these outcomes we designed a structured online survey for the quantitative phase. The survey, completed by 72 respondents, strengthens the interview results by revealing the extent to which the techniques and practices have penetrated the spreadsheet user community. The results show,

1) Spreadsheet testing is common, although done manually for most part, as usage of tools and automation is rare.
2) The handful of six testing techniques in use are lacking in formalism. Among them invariant-based testing is popular, and the only semi-automatic testing technique used in practice.
3) Ensuring correctness of spreadsheets is an important concern, and spreadsheet users believe testing reduces chances of errors.
4) Although testing tools and automation are rarely used, users are welcoming to the notion and believe they would help.
5) Although testing is common, users are not largely satisfied with the quality of their testing, as both critical and non-critical errors often remain even after performing test activities.
6) Apart from testing, spreadsheet users attempt to prevent errors by following a set of development techniques, review processes, and access control.
7) Lastly, maintenance of spreadsheets is important as they are often used for more than 6 or 12 months. Thus, in order to ensure correctness of spreadsheets over longer periods of time, spreadsheet users follow a set of maintenance practices consisting of documentation and manual implementation of version control.

Researchers interested in spreadsheet technology can use these outcomes to deepen their understanding of existing practices in the spreadsheet industry, while looking for opportunities to improve upon. Individuals interested in supporting spreadsheet users with tools can similarly focus on building tools that address the practices already popular, to achieve faster adoption in the industry. Lastly, spreadsheet practitioners can benefit from knowing what practices their peers are following, and start adopting the practices for ensuring correctness of their spreadsheets.

For future work, based on the findings of this study, we firstly aim to investigate in further detail, what the barriers of spreadsheet testing are. We obtained a brief glimpse of that through the rationales provided by the participants for not testing, or not testing extensively (Section IV.C.2.b), discussion of which we left out of scope of this paper. A second opportunity of improvement we recognize is in the technique of invariant-based testing, which is currently practiced in a semi-automatic fashion. We believe it is worthwhile to investigate if this technique can be further automated. Through these research directions, our ultimate goal is to help spreadsheet users with better techniques and tools for testing.

References

[1] A. F. Blackwell, "Psychological issues in end-user programming," in *End user development*. Springer, 2006, pp. 9–30.

[2] F. Hermans, "Analyzing and visualizing spreadsheets," Ph.D. dissertation, Delft University of Technology, 2013.

[3] R. R. Panko, "What we know about spreadsheet errors," *Journal of End User Computing*, vol. 10, pp. 15–21, 1998.

[4] F. Hermans, B. Jansen, S. Roy, E. Aivaloglou, A. Swidan, and D. Hoepelman, "Spreadsheets are code: An overview of software engineering approaches applied to spreadsheets," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 5, March 2016, pp. 56–65.

[5] J. W. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2013.

[6] A. Strauss and J. Corbin, *Basics of qualitative research: Techniques and procedures for developing grounded theory* . Sage Publications, Inc, 1998.

[7] W. C. Hetzel and B. Hetzel, *The complete guide to software testing*. John Wiley & Sons, Inc., 1991.

[8] G. J. Myers, C. Sandler, and T. Badgett, *The art of software testing*. John Wiley & Sons, 2011.

[9] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research: A critical review and guidelines," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16. New York, NY, USA: ACM, 2016, pp. 120–131. [Online]. Available: http://doi.acm.org/10.1145/2884781.2884833

[10] R. Abraham and M. Erwig, "Ucheck: A spreadsheet type checker for end users," *Journal of Visual Languages & Computing*, vol. 18, no. 1, pp. 71–95, 2007.

[11] K. J. Rothermel, C. R. Cook, M. 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*. IEEE, 2000, pp. 230–239.

[12] D. Jannach, T. Schmitz, B. Hofer, and F. Wotawa, "Avoiding, finding and fixing spreadsheet errors–a survey of automated approaches for spreadsheet qa," *Journal of Systems and Software*, vol. 94, pp. 129–150, 2014.

[13] M. Fisher, M. Cao, G. Rothermel, C. R. Cook, and M. M. Burnett, "Automated test case generation for spreadsheets," in *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*. IEEE, 2002, pp. 141–151.

[14] 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)*, vol. 15, no. 2, pp. 150–194, 2006.

[15] R. Abraham and M. Erwig, "Autotest: A tool for automatic test case generation in spreadsheets," in *Visual Languages and Human-Centric Computing (VL/HCC'06)*. IEEE, 2006, pp. 43–50.

[16] M. Burnett, C. Cook, O. Pendse, G. Rothermel, J. Summet, and C. Wallace, "End-user software engineering with assertions in the spreadsheet paradigm," in *Proc. of ICSE '03*, 2003, pp. 93–103. [Online]. Available: http://dl.acm.org/citation.cfm?id=776816.776828

[17] S. E. Kruck, "Testing spreadsheet accuracy theory," *Information & Software Technology*, vol. 48, no. 3, pp. 204–213, 2006.

[18] K. McDaid, A. Rust, and B. Bishop, "Test-driven development: can it work for spreadsheets?" in *Proceedings of the 4th international workshop on End-user software engineering*. ACM, 2008, pp. 25–29.

[19] R. R. Panko, "Recommended practices for spreadsheet testing," *arXiv preprint arXiv:0712.0109*, 2007.

[20] L. Pryor, "When, why and how to test spreadsheets," *arXiv preprint arXiv:0807.3187*, 2008.

[21] F. Hermans, "Improving spreadsheet test practices," in *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research*. IBM Corp., 2013, pp. 56–69.

TUD-SERG-2017-002
ISSN 1872-5392