A Quality Model for Spreadsheets

Jácome Cunha, João Paulo Fernandes, Christophe Peixoto, João Saraiva
HASLab / INESC TEC & Universidade do Minho, Portugal
{jacome,jpaulo,christophe,jas}@di.uminho.pt

***Abstract*—In this paper we present a quality model for spreadsheets based on the ISO/IEC 9126 standard that defines a generic quality model for software. To each of the software characteristics defined in the ISO/IEC 9126, we associate an equivalent spreadsheet characteristic. Then, we propose a set of spreadsheet specific metrics to assess the quality of a spreadsheet in each of the defined characteristics. To obtain the normal distribution of expected values for a spreadsheet in each of the proposed metrics, we have executed them in the widely used EUSES spreadsheet corpus. Then, we quantify each characteristic of our quality model after computing the values of our metrics, and we define quality scores for the different ranges of values. Finally, to automate the quality assessment of a given spreadsheet, according to our quality model, we have integrated the computation of the metrics it includes in both a batch and a web-based tool.**

I. Introduction

Spreadsheets are widely used both by professional and non-professional programmers. Non-professional programmers see them as simple, flexible and easy-to-use calculators, whereas professional ones exploit them in software intensive organizations as advanced intermediate representations to perform data migration between software systems and to perform operations to enrich or simplify data, for example.

The simplicity of spreadsheets and their attractive visual representation has ensured them a great popularity, being nowadays one of the most used software systems in the world. Unfortunately, spreadsheets are known for being error prone, as reported by numerous studies which show that up to 90% of real-world spreadsheets contain errors [1], [2]. This amount of errors suggests that the overall quality of spreadsheets is inherently low, which has motivated the development of approaches for error detection and prevention in them [3], [4]. However, and surprisingly, with the exception of the works [5], [6] and [7], little work has been done on trying to assess the quality of general purpose spreadsheets. As a result, there are no methods, techniques nor tools to measure and, consequently, to improve the quality of a spreadsheet.

In this paper we propose a quality model for spreadsheets. This model is defined by introducing a set of domain specific metrics for spreadsheets which are used to measure concrete spreadsheet characteristics. Such characteristics are related to the general notions of software quality as defined on the ISO/IEC 9126 standard [8]. To calibrate our quality model we use a large repository of real-world spreadsheet instances, namely the EUSES spreadsheet corpus [9]. To automate the assessment of the quality of a spreadsheet, we have implemented our quality model in a (batch) software tool which, together with the EUSES corpus, allows us to automatically study the behavior of the spreadsheet metrics, to give quality scores to spreadsheets, and to evolve our quality model.

This paper is organized as follows: in section II we briefly present the ISO/IEC 9126 software quality standard and how we instantiate it to the spreadsheet realm. In section III we build on the ISO/IEC 9126 to propose a spreadsheet quality model. The calibration of the proposed model is discussed in section IV, and the model itself is evaluated in section V. In section VI we compare our work with works whose goals are related to ours and in section VII we conclude the paper.

II. A Quality Model for Spreadsheets

We define a quality model for spreadsheets based on the widely accepted ISO/IEC 9126 international standard for software product quality [8]. This standard provides a terminology regarding the concept of software product quality that distinguishes six main characteristics. These characteristics are then sub-divided in sub-characteristics as shown in Figure 1.

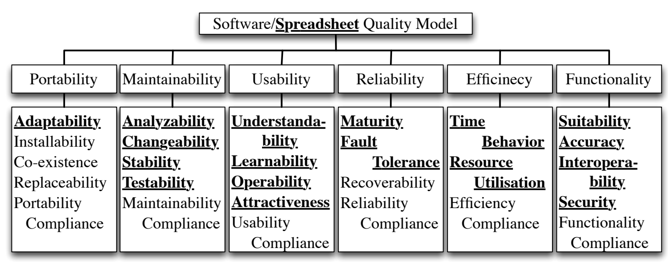

Fig. 1: The ISO/IEC 9126 standard for software quality.

In order to define our quality model for spreadsheets we follow a three phase approach:

1) Firstly, we instantiate the generic ISO/IEC 9126 characteristics to the context of spreadsheets. That is to say that we define domain specific spreadsheet characteristics that directly relate to the more general ISO/IEC 9126 characteristics. Because the ISO/IEC 9126 was created with the goal of assessing the quality of all software products, there are some characteristics of the model that do not directly apply to spreadsheets. This is the case, for

This work is funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects FCOMP-01-0124-FEDER-010048 and FCOMP-01-0124-FEDER-020484. The three first authors were supported by FCT grants SFRH/BPD/73358/2010, SFRH/BPD/46987/2008 and BI3-2011PTDC/EIA-CCO/108613/2008, respectively.

example, of the *recoverability* characteristic that is very important in most software products, but not in regular spreadsheet systems. In Figure 1 the sub-characteristics shown in bold and underlined are the ones we directly map into spreadsheet equivalents.
Moreover, we define a set of spreadsheet metrics that allow us to assess the quality of those characteristics.

2) Secondly, in order to calibrate our quality model we consider a large repository of real-world spreadsheets: the EUSES spreadsheet corpus – a shared resource for supporting experimentation with spreadsheets [9]. The calibration of the quality model is then performed by computing the metrics for all spreadsheets in the repository. Then, and based on this global analysis of the entire repository, we define for each metric used in each characteristic, six intervals of possible metric values. These intervals are used to define a five star ranking as it is currently widely used on software product markets. Moreover, by combining different characteristic scores/stars we are able to give quality scores/stars to spreadsheets.
3) Thirdly, to evaluate and further calibrate our quality model we manually assess the quality of several EUSES spreadsheets, and we compare these results to the automatically generated ones. This manual evaluation allowed us to evolve our quality model.

The calibration of our model requires that we process a large number of spreadsheets. Thus, we have developed a tool that computes several metrics given a spreadsheet. The three phases of our approach are described in the next three sections.

III. The ISO/IEC 9126 for Spreadsheets

In this section we discuss each of the ISO/IEC 9126 characteristics in the context of spreadsheets, and we discuss spreadsheet metrics we may use to assess the quality of such characteristics. In fact, we discuss internal product metrics only which may not be indicators for the external product quality factors, as listed in the ISO/IEC 9126.

Functionality: Functionality is the capacity of spreadsheets to satisfy the needs of their end users, either implied or stated. It is divided by the following four sub-characteristics:

1) *Suitability*: is the quality of a spreadsheet having the right properties for a specific purpose. To assess the quality of this characteristic we consider that if a spreadsheet has formulas with references to blank cells then it does not fit the desired property. Similarly, spreadsheet incongruences may also affect *suitability*. By a spreadsheet incongruence we mean, for example, a cell whose type deviates from a pattern: when all cells in a column are defined by a consistent formula, except one which is defined by a constant value.
2) *Accuracy*: is the faithful measurement or representation of the correctness of a spreadsheet. A great number of both incongruences and blank cells referenced in formulas reduce the accuracy of a spreadsheet. In the same line, if a spreadsheet has many output cells containing errors, then its accuracy will be greatly affected.
3) *Interoperability*: is the ability of two or more worksheets to exchange information and to use the information that has been exchanged. We can say that a spreadsheet has a good interoperability if most of its formulas are correct since that means that the components are changing information without any problem. If the spreadsheet has many references, cells with references and data been exchanged between its worksheets then it has a good interoperability.
4) *Security*: is the ability of a spreadsheet to be protected. Since spreadsheets are mostly used by end users with little knowledge of programming, a spreadsheet is safe if some of its cells are protected, preventing the user to change them by mistake. Besides that, the spreadsheet security can be increased with a password locking the workbook or the worksheets, preventing anyone unauthorized to access or change it.

Reliability: Reliability is the capacity of a spreadsheet to maintain its level of performance under stated conditions for a stated period of time. It is divided in the following two sub-characteristics:

1) *Maturity*: The state of quality of a spreadsheet being fully developed. If the spreadsheet contains non used worksheets, then we consider that it may not be completely developed. The existence of empty labels in rows or columns, blank cells, or tables with blank cells may also indicate that the spreadsheet is under development.
2) *Fault Tolerance:* It is the property of a spreadsheet to continue operating properly in the event of one or more faults within some of its components. If a cell is referenced by many other cells (directly or indirectly), lesser is its fault tolerance since modifying that cell can diffuse mistakes on all the cells that reference it. Furthermore, if a spreadsheet contains many formulas that are nontrivial, it is less fault tolerant because those formulas may reference many cells, and changing just one such cell can lead to an error on the formula result.

Usability: Usability is the capacity of the spreadsheet to be understood, learning how it works, be used and intuitive to the end user. It is divided by the following four sub-characteristics:

1) *Understandability:* It is capacity of being understood. There are many ways to make a spreadsheet more understandable, being the most important the separation of the input, computation and output components of the spreadsheet. Furthermore, if we use different background colors for different types of data cells, then it is likely that a spreadsheet becomes easier to understand. Finally, it is often the case that having a large number of cells makes a spreadsheet hard to understand.
2) *Learnability:* It is the capacity to enable end users to use the spreadsheet. The more cells and complex formulas a spreadsheet has, more difficult it is to learn how to use it. The same occurs when the spreadsheet has many

references and data is interchanged among its worksheets. Improving *learnability* may also be achieved, for example, by giving different background colors for different types of data, and by separating the input, the computation and the output parts of the spreadsheet.

3) *Operability:* It is the capacity of the spreadsheet to be operated. If we follow a modular approach to spreadsheet programming, by separating the input from both the computation and output then the spreadsheet becomes easier to operate. If the spreadsheet has data validation drop down lists, it is also easier to operate.
4) *Attractiveness:* It is the capacity of a spreadsheet to be visually attractive, or appealing. As it occurs with other sub-characteristics of *usability*, it is important to have the input, the computation and the output parts of the spreadsheets separated from each other. Besides that, by giving different background colors for different types of data cells and having data validation drop down lists may also improve the attractiveness of a spreadsheet.

Efficiency: Efficiency measures if the amount of resources used is compatible with the performance level of a spreadsheet. It is divided in the following two sub-categories:

1) *Time Behavior:* Both search formulas and `vlookup` functions are operations that may require intensive computation, so the use of such operations can influence the time behavior of a spreadsheet. Moreover, the number of complex formulas can also affect the time behavior.
2) *Resources Utilization:* Since the `vlookup` function uses many resources, the number of such functions influences the resources needed. Furthermore, if the spreadsheet has a high number of non-blank cells or complex formulas, it is going to use more resources.

Maintainability: Maintainability is the capacity of a spreadsheet to be modified, either to expand functionalities or to correct errors. It is divided in four sub-characteristics:

1) *Analyzability:* It is the capacity of a spreadsheet to be analyzed and the effort needed to diagnose deficiencies in it. If a spreadsheet has a big number of cells, references or formulas, then it is harder to analyze and to find its deficiencies. A good way to make a spreadsheet easier to analyze is to have the data well organized by separating its input, computation and output parts.
2) *Changeability:* Changeability measures how well a spreadsheet can be changed, the effort that is needed to apply modifications in it, and how easy it is to remove its faults. A spreadsheet with well-organized data is easier to change and to remove faulty parts. On the other hand, if the spreadsheet has a great number of cells or references, then it is hard to change.
3) *Stability:* It is the capacity to be stable. A spreadsheet loses stability if it has a big number of cells been referenced by other cells, since erroneously changing that first cell can spread errors all over the spreadsheet. It also loose stability if it has many complex formulas, since changing just one cell from the many referenced by the complex formula can change the formula result into a wrong one.
4) *Testability:* It defines how well a spreadsheet can be tested. The relevance on testing is to confirm that the results one has are correct. On spreadsheets we need to check the output cells, which can be formula cells or data cells, but the last ones are usually labels, and so we just need to test the formula cells. A bigger number of formula cells makes it hard to test the spreadsheet.

Portability: Is the capacity of being transferred from one environment to another. It consists of a single sub-characteristic:

1) *Adaptability:* It defines how well can the spreadsheet be adapted to environmental changes. Since macros are not compatible with all environments or spreadsheet applications, the higher number of macros on a spreadsheet, the less adaptable that spreadsheet is.

IV. Using Metrics to Calibrate the Model

In the previous section we have defined the ISO/IEC 9126 characteristics in the context of spreadsheets. We have also discussed spreadsheet metrics which can be used to assess quality of those characteristics. In this section we study how to calibrate our quality model based on spreadsheet metrics.

To calibrate our model we need to consider a large repository of spreadsheets. The goal is to compute the metrics in a large set of real-word spreadsheet in order to define metric values that allow us to assign a five star ranking to a spreadsheet. Thus, we consider the large EUSES spreadsheet corpus that consists of 5607 spreadsheet files, classified according to six categories, namely, financial (19% of the total number of spreadsheets), inventory (17%), homework (14%), grades (15%), database (17%) and modeling (17%) (the remaining 1% represents other spreadsheets).

The calibration of the quality model is then performed by computing the metrics for all spreadsheets in the EUSES repository. Then, and based on this global analysis of the complete repository, we define for each metric used in each characteristic, six intervals of possible metric values. There are several techniques used to aggregate metrics to ratings [5]. In our quality model we consider the central tendency using the median as suggested in [10]. Figure 2 displays the five graphics that represent the aggregation based on the central tendency of the results of computing the metric “number of cells”, for the complete EUSES spreadsheet repository, in a five level rating.

The third graph (reading from the left to the right, top down) represents the metric “number of cells” to all the spreadsheet in the EUSES. After calculating the median of the number of cells, we represent the spreadsheets with values below it in the fourth graph. Again, we calculate the median for these values and represent the ones below it in the fifth graph. Finally, the median of the last set of values in the last graph gives us the first interval, [0, 208]. The symmetric is done to the values above the median. The second interval is]208, 385] where the upper bound is the median of the fourth graph. The third interval is]385, 1045], where 1045 is the median of the third graph. The fourth interval is]1045, 2858], where the upper

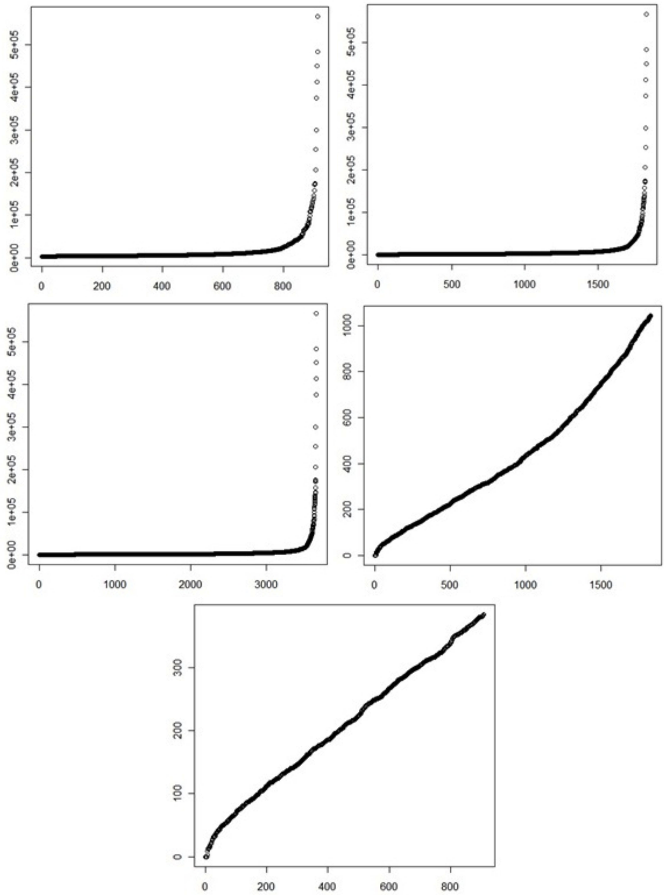

Fig. 2: Graphics of the analysis of the metric "number of cells". A point (x, y) represents the x^{th} spreadsheet with the smallest number of cells, which has exactly y cells.

bound is the median of the second graph. The fifth interval is]2858, 5794], where the upper bound is the maximum number of cells existing in a spreadsheet.

This process has been performed on all the metrics we consider, and the intervals are used to define a 5 star ranking, where a 0(5) star classification represent the lowest(highest) quality ranking, as it is widely used on software product markets. Table I presents for each characteristic of our quality model, the intervals of metric values defining 5 star ratings.

V. Model Evaluation

In this section we present the evaluation we performed to assess the sharpness of the results produced by our model.

We have implemented our quality model in two different tools: a batch and a web-based spreadsheet quality assessment tool. The batch tool can be used to analyze regular spreadsheets and the web-based version to assess the quality of spreadsheets within the GoogleDocs environment.

For each EUSES category, we have randomly chosen five spreadsheets whose quality was determined automatically by our tool. In Figure 3 we present the results obtained when assessing the quality of five spreadsheets in the *database* category: for each characteristic considered, a star-based mark is assigned.

Characteristics - Files	08_content_inventory	Budget	CollectionsReports	Basicdata	centermaterials
Functionality	★★★★☆	★★★☆☆	★★★★☆	★★★☆☆	★★★★☆
Suitability	★★★☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	★★★★☆
Accuracy	★★★☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	★★★★☆
Interoperability	★★★★★	★★★★☆	★★★★★	★★★★★	★★★★★
Reliablity	★★★★☆	★★★☆☆	★★★★★	★★★★★	★★★★☆
Maturity	★★★☆☆	★★☆☆☆	★★★★☆	★★★★☆	★★★★☆
Fault Tolerance	★★★★★	★★★★☆	★★★★★	★★★★★	★★★★★
Usability	★★★☆☆	★★☆☆☆	★★★★☆	★★★☆☆	★★★★☆
Understanbility	★☆☆☆☆	☆☆☆☆☆	★★★☆☆	★★★☆☆	★★★☆☆
Learnability	★★★★☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆
Operability	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★
Actractiveness	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆
Efficiency	★★★★★	★★★★☆	★★★★★	★★★★★	★★★★★
Time behavior	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★
Resource utilisation	★★★★☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆
Maintainability	★★★★☆	★★★☆☆	★★★★★	★★★★★	★★★★★
Analysability	★★★★☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆
Changeability	★★★☆☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆
Stability	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★
Testability	★★★★★	★★★☆☆	★★★★★	★★★★★	★★★★★

Fig. 3: Quality evaluation of EUSES (*database*) spreadsheets.

Using our batch tool we were able to assess the quality of the complete EUSES. In order to assess the appropriateness of our tool (and of our model) we have manually inspected 30 EUSES spreadsheets and we have compared the grades we would manually attribute to a spreadsheet against the grades automatically calculated by our tool. We report in the next paragraphs the results of this exhaustive work; our analysis follows the categories defined within EUSES.

a) Database Category: The sub-characteristics *suitability* and *accuracy* of *functionality* were not always correctly computed. However, a manual analysis revealed that one of the analyzed spreadsheets was being used as a text document and not as a spreadsheet, which led to imprecise automatic results. Nevertheless, we have not found any other similar occurrence. The *maturity* of *reliability* were also inappropriately calculated for two spreadsheets. This was due to the fact the algorithm that calculates the empty-cells metric considers as empty cells the rows/columns with no data separating different parts of the spreadsheet. In fact, this is a problem reported in other contexts [11]. In this case, the algorithm should be improved to discard the empty cells that are used to improve the layout of the spreadsheet. Finally, the *attractiveness* of *usability* was also calculated incorrectly for one spreadsheet. The reason for this was that the current metrics do not give information about the organization of the data, that is, its *layout* (e.g., different colors in the labels). We believe, however, that to improve the usability characteristic star rating requires extremely complex metrics, and this is a direction that we have not yet explored.

b) Modeling Category: The *maturity* sub-characteristic of *reliability* was miss calculated for one spreadsheet. The reason remains the same as before: the empty-cells metric can still be improved. The *understandability* of *usability*, for one spreadsheet was imprecisely rated, again due to the nature of the empty-cells metric. Finally, the *understandability* and *attractiveness* of *usability* were imprecisely calculated because we have not considered layout metrics.

c) Homework Category: The *maturity* sub-characteristic of *reliability* was miss calculated for one spreadsheet because of the empty-cells metric algorithm. The *understandability*

Metric	5 stars	4 stars	3 stars	2 stars	1 stars	0 stars
Functionality						
Suitability						
number of incongruences	[0,2]	]2,9]	]9,18]	]18,62]	]62,141]	>141
number of blank cells referenced in formulas	[0,4]	]4,12]	]12,55]	]55,266]	]266,710]	>710
Accuracy						
number of incongruences	[0,2]	]2,9]	]9,18]	]18,62]	]62,141]	>141
number of blank cells referenced in formulas	[0,4]	]4,12]	]12,55]	]55,266]	]266,710]	>710
number of output cells with errors/bad content						
Interoperability						
number of cells that have references	[0,2]	]2,12]	]12,63]	63,187]	]187,416.5]	>416.5
number of references	[0,33]	]33,80]	]80,304]	]304,1141]	]1141,2974]	>2975
data been exchanged between sheets; number of correct formulas						
Security						
number of protected cells for writing; password to lock data; hidden information; data validity constraints						
Reliability						
Maturity						
difference between the nr. of existing sheets and the nr. of sheets used	0	1	2	]2,9]	]9,15]	>15
number of blank cells	[0,8]	]8,60]	]60,320]	]320,1271.5]	]1271.5,3341]	>3341
number of labeled rows/columns that are empty						
Fault Tolerance						
number of referenced cells	[0,33]	]33,80]	]80,304]	]304,1141]	]1141,2974]	>2975
number of complex formulas	[0,1]	]1,4]	]4,18]	]18,66]	]66,162]	>162
Usability						
Understanbility						
number of cells	[0,208]	]208,385]	]385,1045]	]1045,2858]	]2858,5794]	>5794
different colors for different types of data; separate input, computation and output						
Learnability						
number of complex formulas	[0,1]	]1,4]	]4,18]	]18,66]	]66,162]	>162
number of cells	[0,208]	]208,385]	]385,1045]	]1045,2858]	]2858,5794]	>5794
number of references	[0,33]	]33,80]	]80,304]	]304,1141]	]1141,2974]	>2974
different colors for different types of data; separate input, computation and output; amount of data being exchanged between sheets						
Operability						
number of referenced cells	[0,2]	]2,12]	]12,63]	]63,187]	]187,416.5]	>416.5
create/have data validation drop down lists; separate input, computation and output						
Attractiveness						
number of non-blank columns	[0,3]	]3,5]	]5,8]	]8,12]	]12,18]	>18
number of cells	[0,208]	]208,385]	]385,1045]	]1045,2858]	]2858,5794]	>5794
create/have data validation drop down lists; different colors for different types of data; separate input, computation and output						
Efficiency						
Time Behavior						
number of complex formula	[0,1]	]1,4]	]4,18]	]18,66]	]66,162]	>162
Time Behavior						
number of non-blank cells	[0,81]	]81,162]	]162,412]	]412,1125]	]1125,2196]	>2196
number of formulas	[0,5]	]5,16]	]16,65]	]65,235]	]235,540]	>540
Maintainability						
Analyzability						
number of cells	[0,208]	]208,385]	]385,1045]	]1045,2858]	]2858,5794]	>5794
number of formulas	[0,5]	]5,16]	]16,65]	]65,235]	]235,540]	>540
number of references	[0,33]	]33,80]	]80,304]	]304,1141]	]1141,2974]	>2974
data organization						
Changeability						
number of cells	[0,208]	]208,385]	]385,1045]	]1045,2858]	]2858,5794]	>5794
number of referenced cells	[0,2]	]2,12]	]12,63]	]63,187]	]187,416.5]	>416.5
data organization						
Stability						
number of complex formulas	[0,1]	]1,4]	]4,18]	]18,66]	]66,162]	>162
number of referenced cells	[0,2]	]2,12]	]12,63]	]63,187]	]187,416.5]	>416.5
Testability						
number of formulas	[0,5]	]5,16]	]16,65]	]65,235]	]235,540]	>540
Portability						
Adaptability						
quantity of macros code						

TABLE I: Calibration of the quality model based on the metrics calculated on the EUSES spreadsheets.

and *attractiveness* of *usability*, for one spreadsheet, were also inaccurately calculated because no layout information is calculated.

d) Inventory Category: The *understandability* and *attractiveness* sub-characteristics of *usability*, for one spreadsheet were imprecisely calculated again because no layout information is exploited.

For the remaining (sub-)characteristics we have observed that the computed stars are in fact precise and accurate.

For each spreadsheet a total of 20 ratings (number of stars) is calculated (only for sub-characteristics and characteristics). Thus, for the 30 spreadsheets analyzed, our tool computed 600 ratings. From our manual inspection, we have only observed 24 imprecise star-ratings. This means that our tool rated accurately 96% of the characteristics and sub-characteristics. From this exhaustive analysis, we believe it is appropriate to conclude that in general our tool is producing good results. Nevertheless, we have detected some limitations in our model:

- In spreadsheets, empty cells are very often used with layout purposes (very much as spaces are used in regular programming languages). The software used to compute the empty cells metric does not handle layout information, so such "layout" empty cells are considered!
  We may use Erwig's spacial logic algorithms to have a more precise metric for empty cells [12], [13].
- Our software does not consider the use of colors, tables, and other layout information. Thus, more non-trivial metrics must be added to the algorithm so more precise quality ratings can be calculated.

This is the very first iteration in the goal of defining a quality model for spreadsheets. This model needs now to be evolved, both by adding new metrics to the model and by defining new intervals for the star ranking. In fact, from our evaluation, we can conclude that our spreadsheet quality model can be improved: the sub-characteristics *understandability* and *attractiveness* should be enhanced with a new metric to calculate the quality of the spreadsheet layout. This would improve the quality of the rating computed by our model to the *usability* characteristic, which is the one that was more often miss-calculated.

VI. Related Work

In [6] the authors take a first step towards automated assessment of spreadsheet maintainability. As in the work here presented, they apply the selected metrics to the EUSES spreadsheet corpus in order to study their behavior. Their work, however, is restricted to achieve a maintainability model whilst we defined a first complete spreadsheet quality model.

In [14] the authors sketch a new maintainability model that alleviates some problems reported by other techniques. Since this work was done in an industrial environment, the authors discuss their experiences with using such a model for IT management consultancy activities. Although their model features only maintainability of software, their technique to calibrate the model is similar to the one we used.

Metrics for spreadsheets have already been defined [15], [16]. Nevertheless, to the best of our knowledge, our work is the very first attempt to define a complete spreadsheet quality model. Although this is a first proposal, we have already defined it based on the general software quality model, calibrated it based on the most widely used (in science) set of spreadsheets, and manually validated it, having this validation suggested some improvements. Nevertheless, an independent validation would give more support to the model. Thus, we plan to revalidate the results here presented with external professional spreadsheet users.

VII. Conclusions

This paper presented a quality model for spreadsheets based on the ISO/IEC 9126 standard. Spreadsheet characteristics were considered to model the ISO/IEC 9126 standard, and spreadsheet metrics were defined to assess the quality of such characteristics. We calibrated and validated our quality model using the large EUSES spreadsheet repository. Finally, we implemented our quality model in a software tool and our first experimental results show the tool is able to give good results when automatically assessing the quality of a spreadsheet.

References

[1] R. R. Panko, "Spreadsheet errors: What we know. what we think we can do," *Symp. of the European Spreadsheet Risks Interest Group (EuSpRIG)*, 2000.

[2] R. Panko, "Facing the problem of spreadsheet errors," *Decision Line, 37(5)*, 2006.

[3] R. Abraham, M. Erwig, S. Kollmansberger, and E. Seifert, "Visual Specifications of Correct Spreadsheets," in *Symp. on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE CS, 189–196, 2005.

[4] J. Cunha, J. P. Fernandes, J. Mendes, and J. Saraiva, "MDSheet: A framework for model-driven spreadsheet engineering," in *Int. Conference on Software Engineering (ICSE)*. ACM, 2012, pp. 1395–1398.

[5] J. P. Correia and M. A. Ferreira, "Requirements for automated assessment of spreadsheet maintainability," *CoRR*, vol. abs/1111.6902, 2011.

[6] ——, "Measuring maintainability of spreadsheets in the wild," in *IEEE 27th Int. Conf. on Software Maintenance (ICSM)*, 2011, pp. 516–519.

[7] A. v. D. Felienne Hermans, Martin Pinzger, "Measuring spreadsheet formula understandability," in *Symp. of the European Spreadsheet Risks Interest Group (EuSpRIG)*, 2012, to appear.

[8] ISO(2001), "ISO/IEC 9126-1: Software engineering - product quality - part 1: Quality model," Geneva, Switzerland.

[9] M. F. II and G. Rothermel, "The EUSES Spreadsheet Corpus: A shared resource for supporting experimentation with spreadsheet dependability mechanisms," in *Workshop on End-User Software Engineering*, 2005, pp. 47–51.

[10] D. Spinellis, "A tale of four kernels," in *International conference on Software engineering (ICSE)*. ACM, 2008, pp. 381–390.

[11] J. Cunha, J. P. Fernandes, H. Ribeiro, and J. Saraiva, "Towards a catalog of spreadsheet smells," in *International Conference on Computational Science and Its Applications (ICCSA)*. LNCS 7336, 2012, pp. 202–216.

[12] R. Abraham and M. Erwig, "Header and unit inference for spreadsheets through spatial analyses," *Symp. on Visual Languages and Human Centric Computing (VL/HCC)*, pp. 165–172, 2004.

[13] ——, "UCheck: A spreadsheet type checker for end users." *Journal of Visual Languages and Computing*, vol. 18, no. 1, pp. 71–95, 2007.

[14] I. Heitlager, T. Kuipers, and J. Visser, "A practical model for measuring maintainability," in *Int. Conference on Quality of Information and Communications Technology (QUATIC)*. IEEE CS, 2007, pp. 30–39.

[15] A. Bregar, "Complexity metrics for spreadsheet models," *CoRR*, vol. abs/0802.3895, 2008.

[16] K. Hodnigg and R. T. Mittermeir, "Metrics-based spreadsheet visualization: Support for focused maintenance," *CoRR, abs/0809.3009, 2008.*