Towards a Method for Evaluating the Precision of Software Measures*

Beatriz Marín[1], Nelly Condori-Fernández[1], Oscar Pastor[1]

[1]*Department of Information Systems and Computation*
Technical University of Valencia
Camino de Vera s/n
46022 Valencia, España
{bmarin, nelly, opastor}@dsic.upv.es

Abstract

Software measurement currently plays a crucial role in software engineering given that the evaluation of software quality depends on the values of the measurements carried out. One important quality attribute is measurement precision. However, this attribute is frequently used indistinctly and confused with accuracy in software measurement. In this paper, we clarify the meaning of precision and propose a method for assessing the precision of software measures in accordance with ISO 5725. This method was used to assess a functional size measurement procedure. A pilot study was designed for the purpose of revealing any deficiencies in the design of our study.

1. Introduction

Quality in software engineering can be related to the process of software production or to the software product. Evaluation of software product quality implies measurement of the quality attributes that we want the software product to have, for instance software product functionality, maintainability, usability, and so on.

The control and management of software product quality will be affected by measurement quality. Consequently, software measures must also have certain quality attributes such as accuracy and precision. Accuracy is related to the closeness to the 'true value' of the measurements [7]. Since the 'true value' is only a theoretical concept, when a measurement is performed it is important to obtain the value for the measurement and the estimation of the degree of uncertainty. However, the degree of uncertainty of a software measurement is very difficult to obtain because external factors exist that affect measurements. One has to first obtain precise measures in order to obtain accuracy measures. In this paper we focus on this important attribute: measurement precision.

In the literature, we have not found a rigorous method for measuring and evaluating precision. The main contribution of this paper is to define a generic method that allows measurement of the precision of software measures. This method was designed in accordance with ISO 5725 [7], a standard which is widely used in other sciences but surprisingly is not used in software engineering.

The rest of the paper is organized as follows: Section 2 presents the research problem. Section 3 presents a method for the evaluation of the precision of software measurements. Section 4 presents a pilot study that illustrates the application of the precision method, and Section 5 presents some lessons learned from the results obtained in the pilot study. Finally, Section 6 presents some conclusions and further work.

2. Research Problem

The ISO 5725 [7] standard - defines precision as the closeness of agreement of test results. This standard presents the formulae for calculating the repeatability and reproducibility of the measures.

The International Standard for Functional Size Measurement ISO 14143 [6] does not specifically define the term precision. This standard only has a note that advises that the term precision should not be used as a synonym for accuracy. However, ISO 14143 presents the definition of repeatability (closeness of the

* This work has been developed with the support of MEC under the project SESAMO TIN2007-62894 and co financed by FEDER.

agreement between the results of successive measurements of the same measurand carried out under the same conditions of measurement) and reproducibility (closeness of the agreement between the results of measurements of the same measurand carried out under changed conditions of measurement) of the results of measurements. This standard provides a brief example of the calculation of repeatability, but does not show the respective formulae. For the measurement of reproducibility, not even a brief example is provided.

In the software engineering community, the term precision is sometimes incorrectly confused with the term accuracy, which is defined as the closeness to the 'true value' of the measurand in [7]. For instance, Kemerer in [3] uses indistinctly the terms accuracy and precision, and also uses the term reliability instead of reproducibility. These terms should not be used indistinctly because they have a different meaning.

Some authors have taken into account the measurement of the reproducibility of software measures, such as Abrahao et al [10] and Condori et al [8], that evaluate the reproducibility of measures of functional size procedures based on IFPUG-FPA and COSMIC respectively. Both approaches use a statistical equation similar to the one proposed by Kemerer [3]. The main disadvantage of this formula is that it uses the average, which is not correct if the results of the measurements are not homogeneous. Other authors as Diab et al [5] affirm that repeatability is assured with the automation of the measurement procedure. Although we agree with this affirmation, it is important to control repeatability and reproducibility from the design of the measurement procedure in order to detect weakness and improve the measurement procedure before its automation.

By evaluating the precision of software measures in terms of repeatability and reproducibility, we can detect the causes that produce variability of measurements, such as: the knowledge of the subjects that perform the measurement task; the legibility of the instrumentation material; the correctness of the explanation of the measurement procedure, etc.

The next section presents a method for evaluating the precision of software measures.

3. A Method for Evaluating the Precision of Software Measures

We adapted ISO 5725 by instantiating this standard with concepts used in software measurement (see Table 1)

Table 1. Instantiation of ISO 5725 with software engineering concepts.

ISO 5725	Software Engineering
Measurement yield	The measurement method must have a continuous scale and must give a single value as the result of the test.
Operator	Subjects that will measure the software artifacts (managers, analysts, designers, etc). These subjects must have knowledge of software engineering and must also be familiar with the use of software artifacts (software products obtained from any phase of the development process).
Test site	The place where the subject will measure the software products.
Equipment	The software artifact to be measured and the instruments to measure this artifact. The instruments to measure a software artifact can have for manual, semi-automatic, or automatic use.
Laboratories	In our field of study, a laboratory is the combination of: subjects, software artifacts, and instruments (i.e. the measurement procedure).
Different levels of the test	These levels can be related to the complexity or size levels of the software artifact that will be measured. Both criterions of levels are representatives for the measurement of the precision.

Our evaluation method comprises three phases, which are shown in Figure 1.

3.1. Definition phase.

The *Characterization of subjects* activity includes the identification of the level of knowledge of the measurement procedure and the software artifacts of the subjects.

The *Characterization of place* activity includes the specification of the size of the place, illumination, movables, and the computers (i.e. OS, RAM, and software installed) used for the measurement.

The *Preparation of instruments* activity corresponds to the preparation of the materials that the subjects must use to perform the measurement exercise. Moreover, if the subjects have no knowledge of the measurement procedure or of the software

artifacts to be measured, then there will be a need for preparation of instruments for training the subjects.

The material for the measurement exercise consists of instructions for performance of the measurement, a set of software artifacts of different levels, and a results sheet for every software artifact to be measured. Each measurement must be carried out at least twice to allow measurement of repeatability. The instructions of the measurement exercise will be the same for all the subjects to allow measurement of reproducibility.

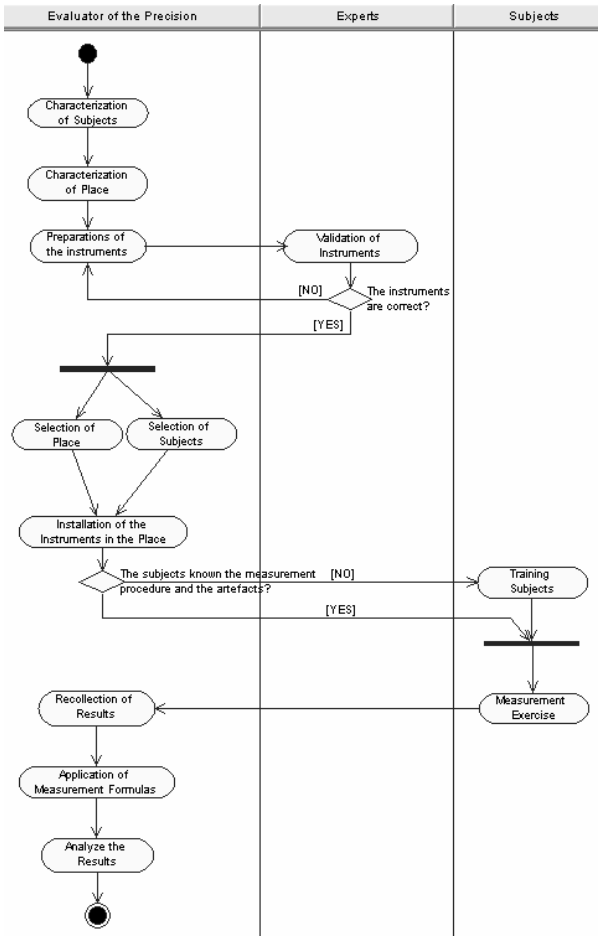

Figure 1. Method for evaluating the precision of software measures.

The *Validation of instruments* activity is carried out by an experts group for revising the software artifacts and the measurement procedure. The number of expert must be small to diminish the noise in the validation.

3.2. Measurement phase.

In the *selection of subjects* activity, the persons that will carry out the measurement exercise must be selected randomly according to the characterization of the subjects. At least a number of 30 subjects would be selected to not to affect to the significance of the measures.

In the *selection of place* activity, the place where the measurement exercise will be carried out must be selected according to the characterization of the place.

The activity of *installation of the instruments in the place* consists of the copy of the prepared instruments in each computer that will used by the subjects. Also this activity includes the print of the instruments that will be given in paper to the subjects.

The *training of subjects* activity is carried out if it is required for the correct understanding of the measurement procedure.

In the *measurement exercise* activity, the subjects perform the measurement following the instructions set out in the definition phase. In this step, the subjects cannot ask questions.

3.3. Evaluation phase.

ISO 5725 has 50 formulas that in conjunction allow calculation of repeatability and reproducibility of the measures and analysis of the values obtained. This standard also has six tables to in which to input data. We selected and adapted six formulae and three tables. This phase comprises of three activities:

The *obtaining results* activity: Each measure obtained (M) by the subjects at each level is recorded at the intersection of subjects and levels of the Table 2. The data obtained must also be validated, by eliminating redundant and missing data; and identifying outliers and outlying subject. The outliers are measures that deviate greatly from comparable entries in Table 2. The outlying subjects are subjects that have several abnormal measures. Then, the evaluator must decide whether the outliers and the outlying subjects are to be ignored or corrected

Table 2. Recording the results of the measures (adapted from [7]).

	Level				
Subject	1	..	J	..	m
1	M 111 M112				M1m1 M1m2
…					
i			… Mijk ….		
…					
n	Mn11 Mn12				Mnm1 Mnm2

When the measures are validated, the cell means must be calculated using Formula 1:

$$\text{Cell}_{ij} = \frac{1}{n_{ij}} \sum_{k=0}^{n_{ij}} Measure_{ijk} \quad (1)$$

Where: $Measure_{ijk}$ is the measure k obtained for a subject i at level j; n_{ij} is the total number of measures carried out in a Cell_{ij}. The cell means calculated are recorded in a table.

Then the spread of each cell must be quantified, by application of Formula 2. The spreads calculated are recorded in a table with levels and subjects.

$$\text{Spread}_{ij} = \sqrt{\frac{1}{n_{ij}-1} \sum_{k=0}^{n_{ij}} (Measure_{ijk} - Cell_{ij})^2} \quad (2)$$

In the *application of measurement formulae* activity the precision is calculated for each level j, using the repeatability variance and the reproducibility variance.

To calculate the repeatability variance (S^2_{rj}), Formula 3 is used.

$$S^2_{rj} = \frac{\sum_{i=1}^{p} (n_{ij} - 1) Spread_{ij}^2}{\sum_{i=1}^{p} (n_{ij} - 1)} \quad (3)$$

Where: p is the number of subjects that perform the measurement exercise.

To calculate the reproducibility variance (S^2_{Lj}), Formula 4 is used.

$$S^2_{Rj} = S^2_{rj} + S^2_{Lj} \quad (4)$$

To obtain the value of S^2_{Lj}, we previously apply Formula 5 for calculating the general mean of a level j represented by m_j. Formula 6 is then applied to obtain the between-subjects variance.

$$m_j = \frac{\sum_{i=1}^{p} n_{ij} Cell_{ij}}{\sum_{i=1}^{p} n_{ij}} \quad (5)$$

$$S^2_{Lj} = \frac{\left(\left(\frac{1}{p-1} \sum_{i=1}^{p} n_{ij} (Cell_{ij} - m_j)^2 \right) - S^2_{rj} \right)}{\frac{1}{p-1} \left[\sum_{i=1}^{p} n_{ij} - \frac{\sum_{i=1}^{p} n^2_{ij}}{\sum_{i=1}^{p} n_{ij}} \right]} \quad (6)$$

Finally, in the *analyze the results* activity, the precision of the measures is analyzed. Low repeatability and low reproducibility indicate high precision. A high repeatability value indicates that the instruments used must be reviewed and rectified or redesigned. On the other hand, a high reproducibility value indicates the possibility that the knowledge of the selected subjects is dissimilar, the measurement procedure has not been understood or better training of the subjects is required.

4. A Pilot Study to Evaluate the Precision of OOmCFP

In this section we will apply our proposal in a pilot study to evaluate the precision of a measurement procedure called OOmCFP [1].

4.1. The OOmCFP Functional Size Measurement Procedure

The OOmCFP procedure [1] was proposed in order to measure the functional size of applications that are automatically generated using the OO-Method approach. OO-Method is a development method based on model transformations that use the MDA principles [9]. OOmCFP focuses on the Conceptual Model of OO-Method (Object Model, Dynamic Model, Functional Model, and Presentation Model).

OOmCFP starts with the definition of the strategy to perform the measurement, which includes: a) The scope of OOmCFP that comprises all the functionality of an OO-Method application, which is specified in the conceptual model of OO-Method; b) The granularity level is low, since all the details in the OO-Method conceptual model are needed to generate the applications; c) The layers identified in the OO-Method applications are the client tier, the server tier, and the database tier; d) The functional users in the OO-Method applications are the human user, the Client tier, the Server tier, and Legacy users that interact (send or receive data) with the layers of the application and also are separated by a boundary of each layer of the OO-Method application.

Once the strategy is defined, OOmCFP starts a mapping phase, where 82 rules were designed to reduce misinterpretation of the generic concepts of COSMIC and facilitate measurement of the functional size of OO-Method applications from their conceptual models. For instance, each class that is used in a functional process is identified as a data group.

The identification of the data movements, a fundamental step in the COSMIC method, OOmCFP has 65 rules to correctly identify these data movements that can be entry (E), exit (X), read (R) or write (W).

For the measurement phase, OOmCFP has 3 measurement rules that allow the quantification of functional size according to the unit defined in COSMIC: 1 CFP (Cosmic Function Point) for each data movement. A complete description of OOmCFP can be visualized in its measurement guide[2].

[2] http://oomethod.dsic.upv.es/labs/images/OOmCFP/guide.pdf

4.2. Applying the Precision Evaluation Method to OOmCFP

The goal of our pilot study, using Goal/Question/Metric template [11], is described as follows: "To analyze the OOmCFP measurement procedure and the instruments used for the measurement exercise for the purpose of evaluating its correctness from the viewpoint of the researcher in the context of Computer Science students measuring OO-Method conceptual models with OOmCFP".

This pilot study has focused on both Definition and Measurement phases, since the evaluation phase is based on selected formulae from ISO 5725 that allow quantifying of software measure precision.

4.2.1. Definition phase. The subjects were characterized as persons with at least some knowledge of the OO-Method conceptual model and no knowledge of the OOmCFP procedure. The place was characterized as a room with sufficient computers for the subjects. The computers must have Windows as Operative System, the Olivanova Modeler tool [2] for the work with the OO-Method conceptual models, and Microsoft Office installed.

The training instruments were the following: a set of slides to teach the main concepts of the OO-Method conceptual model that are used in OOmCFP; a set of slides to teach the OOmCFP procedure; an illustrative example of the application of OOmCFP to the conceptual model of an invoice application; a measurement guide[2]; a results sheet; and the application of OOmCFP to a conceptual model of a Rent a Car application to verify the training process carried out.

The instruments for the measurement exercise were the following: the three conceptual models of OO-Method with three levels of functional size (small, medium, and large), the instructions, and the results sheet. The following conceptual models were used: Publishing application (small-five classes); Photography Agency application (medium - seventeen classes); and Expense Report application (large - twenty three classes).

The instruments were validated by two experts in OO-Method and two experts in OOmCFP. The measurement guide and the results sheet were not well structured, and these instruments had to be changed before the experts could validate all the instruments.

4.2.2. Measurement phase. According to the characterization of the subjects twelve students were selected from the students enrolled in the "Master's Degree in Software Engineering, Formal Methods, and Information Systems" at the Technical University of Valencia from September 2006 to September 2008.

The place selected was the Room 0S02 of the Department of Information Systems and Computation. This room has twenty computers with the programs and instruments described in the definition phase.

Also in this activity the measurement guide and the instructions were printed and located close to each computer of the classroom.

The activity training of subjects was carried out to develop the expertise required to measure the functional size of the conceptual models using OOmCFP. The training method used was the demonstration/practice method [4]. The demonstration part took only 1 hour, and it included the presentation of OO-Method Conceptual Model, the Olivanova Modeler tool, and the OOmCFP procedure. The practice part took 3 hours, and it included the application of OOmCFP to the Rent a Car case study.

It is important to note that 4 students were experts using the Olivanova tool, 5 students had some idea of the use of the tool, and 3 students had never used the tool. The expert students obtained the measurement of the functional size of the Rent a Car application. The students that had notions of the use of the tool carried out the measurement of some functional processes. The students that did not know the tool did not achieve the measurement of any functional process, but did correctly identify the functional process.

Thus, the different levels of knowledge of the tool affected the measurement of the precision; for instance, the experts correctly applied the mapping and measurement rules defined in OOmCFP, but the inexpert students confused the elements of the conceptual model when identifying the data movements. The knowledge of the tool was not taken into account in the characterization of the subjects and the pilot study reflects this is an important factor that affects the measurement of the precision.

In addition, the measurement procedure and the instruments were adjusted because they were not correctly understood by the subjects. Thus the objective of the pilot study carried out was achieved.

5. Lessons Learned from the Results of the Pilot Study for Improvement of OOmCFP

With respect to the OOmCFP procedure:

- When the subjects carried out the measurement many questions arose about how to properly identify each *functional process*. As a result we detailed the rules to identify the functional

processes and the elements that are contained in a functional process.

- When *an inheritance of classes* participates in a functional process, some subjects considered one data group for each class of the inheritance. As a result we included a rule that indicates that when an inherited class participates in a functional process it must be considered as a single data group.
- When the subjects carried out the measurement of the practice model, some rules defined in OOmCFP were never used. We therefore eliminated these rules because tended to confuse the subjects performing the measurement.

With respect to the measurement guide:

- When the subjects identified the data movements, they had difficulties because the rules were organized according to the layers of the OO-Method applications. Thus, the subjects took longer than expected to carry out the measurement exercise. In response, we reorganized the rules for identification of data movements in accordance with the conceptual elements involved in the functional processes.

With respect to the results sheet:

- When the subjects entered the functional processes and the elements contained in each functional process, we note that the subjects had difficulties when the elements had different levels of abstraction in the OO-Method conceptual model. We changed the results sheet in order to differentiate the elements of each level of abstraction that comprise the functional process.

With respect to the training duration:

- The measurement of the conceptual model used in the practice part of the training was intended to take 50 minutes. However, as some subjects took two hours, we simplified the model used for the practical part.

6. Conclusions and Further Work

We have presented a rigorous method for evaluating the precision of software measures based on ISO 5725. We selected only that sub-set of the formulae presented in ISO 5725 most appropriate for software measurement. The method for the evaluation of precision comprised three phases: definition, measurement, and evaluation.

The proposed method quantified precision by calculating repeatability and reproducibility of measures. It attempted to control all the factors that could affect evaluation of the precision of the measures (knowledge of the measurement procedure, experience in using the measurement procedure, etc.). The use of this method helps to evaluate precision in the design phase of the measurements procedure, reducing the cost of the modification of the measurement procedure when it is already automated.

We have carried out a pilot study to evaluate the precision of the OOmCFP functional size measurement procedure. The results obtained in the pilot study suggest the improvement of the OOmCFP procedure and instruments (measurement guide and results sheet). Further work includes carrying out an empirical evaluation of the precision of OOmCFP and other measurement procedures. Further work would also include development of a tool to automate some parts of the method for the evaluation of precision.

7. References

[1] B. Marín, N. Condori-Fernández, O. Pastor and A. Abran, "Measuring the Functional Size of Conceptual Models in a MDA Environment", Accepted in the 20th CAiSE, Montpellier, France, 2008.

[2] CARE Technologies Web Site, www.care-t.com

[3] C.F. Kemerer, "Reliability of Function Points Measurement", Communications of the ACM, 36 (2), 1993, pp. 85-97.

[4] DOE Handbook, Alternative Systematic Approaches to Training, 1995, pp. 1074-1095.

[5] H. Diab, F. Koukane, M. Frappier and R. St-Denis, 2004, "μcROSE: Automated Measurement of COSMIC-FFP for Rational Rose Real Time", Information and Software Technology, 47(3), 2005, pp. 151-166.

[6] ISO, "ISO/IEC 14143-3 – Information Technology – Software measurement – Functional size measurement – Verification of functional size measurement methods", 2003.

[7] ISO, "ISO 5725-2 – Accuracy (trueness and precision) of Measurements Methods and Results – Part 2: Basic Method for the Determination of the Repeatability and Reproducibility of a Standard Measurement Method", 1994.

[8] N. Condori-Fernández, and O. Pastor, "Evaluating the Productivity and Reproducibility of a Measurement Procedure", Proceedings of the 2nd QoIS, 2006, pp: 352-361.

[9] O. Pastor and J.C. Molina, Model-Driven Architecture in Practice, Springer, 2007.

[10] S. Abrahao, G. Poels, O. Pastor, "Assessing the Reproducibility and Accuracy of Functional Size Measurement Methods through Experimentation", ISESE, 2004, pp. 189-198.

[11] V. Basili and H. Rombach, "The TAME Project: Towards Improvement Oriented Software Environments", IEEE Transactions on Software Engineering, 1988, pp. 758-773.