From Software Metrics to Software Measurement Methods: A Process Model

Jean-Philippe Jacquet
E.mail : c3401@er.uqam.ca
Tel : (514) 987-3000 (6667)
Fax : (514) 987-8477

Alain Abran
E.mail : abran.alain@uqam.ca
Tel : (514) 987-3000(8900)
Fax : (514) 987-8477

Research Lab. in Software Engineering Management,
Department of Computer Science,
Université du Québec à Montréal,
P.O. Box 8888, Succ. Centre-Ville,
Montréal (Québec), Canada H3C 3P8

Abstract *This paper presents a process model for software measurement methods. The proposed model details the distinct steps from the design of a measurement method, to its application, then to the analysis of its measurement results and last to the exploitation of these results in subsequent models, such as in quality and estimation models. From this model, a validation framework can be designed for analyzing whether or not a software metrics could qualify as a measurement method. The model can also be used for analyzing the coverage of the validation methods proposed for software metrics.*

Keywords : Software Measurement Methods, Software Metrics, Process Models, Metrics Validation.

1. Introduction

Software measurement will need to play an increasingly important role in software engineering if this discipline is to truly become an engineering discipline. Over the past twenty years, a significant number of software metrics have been proposed to better control and understand software development practices and products. Unfortunately, very few of them have been looked at closely from a measurement method perspective and it is currently difficult, because of a lack of agreed-upon validation procedures, to even analyze the quality of these metrics.

What is a valid metrics, or even a valid measurement method? How do you validate a measurement method? Various authors have attempted to address these questions, but usually from different points of view (mathematical, practical, etc.) in the past few years [2, 4, 8, 9, 11, 18]. For example, Schneidewind proposes a metrics validation methodology based on six validity criteria [18], while other authors [4, 8] postulate that "*a measurement method is valid if it can be shown that it gives a proper numerical characterization of some attributes.*" Nevertheless, this definition is far from being unanimously accepted, and Fenton remarks that "*there is an implicit assumption in the software community that validation of a measurement method in the measurement theory* *sense is not sufficient. Specially it is expected that validation must also entail demonstration that the measure is itself part of a valid prediction system* [8].

Unfortunately, such a complete and practical validation framework does not exist at this time. Kitchenham et al. remark, "*what has been missing so far is a proper discussion of relationships among the different approaches*" [11]. Indeed, they have recognized the need for a validation framework which would take into account, and integrate, the distinct validation perspectives proposed by these various authors, and they have put forward a proposal for such a validation framework and "*encourage researchers and practitioners to respond critically*" [10]. The focus of the validation framework proposed in [11] is based on a definition of a measurement structure model and they analyze the proposed elementary components of measures extensively. The analytical perspective proposed in this paper is complementary and discusses this issue from a measurement method point of view.

Similar questions will need to be investigated for the development of a validation framework of measurement methods, such as :

- Is the measurement method internally valid, i.e. in the sense that it can be shown that it gives a proper numerical characterization of the attribute to be measured?
- Is the measurement method usable? A measurement as perfect as possible from a mathematical view point wouldn't be of any interest if it was not possible to apply it (because it is far too time-consuming, for example).
- Can the measurement method take its place in a valid prediction system?

This paper presents a work-in-progress the aim of which is to clarify, from a validation perspective, what a measurement method consists of and what the different steps are from its conception to its use. In order to do this, section 2 presents a high-level model of a measurement process, and section 3 a detailed model, and the components of this model. Section 4 illustrates how a good understanding of measures from a measurement method point of view will help to validate proposals of software metrics.

It is worth noting that this validation framework is being used to strengthen current work of working group WG12 of the ISO subcommittee on software engineering (ISO-IEC/JTC1/SC7). This ISO working group is, at the present time, working on a proposal to establish an international standard on functional size measurement methods. This proposal is being designed by several expert groups from all over the world (Australia, UK, France, Canada, USA, Germany, etc.). The Research Lab. on Software Engineering Management of the University of Quebec in Montreal supports the Canadian editor of the part of this proposal that deals with the design of a verification guide the aim of which is to evaluate functional size measurement methods that will be proposed for recognition as international standards.

2. Measurement Process: High-level Model

In the literature, in software engineering as well as in psychology or physics, the words (nouns) "measure" and "measurement" are used in different ways:
- to refer to a method allowing the assignment of a numerical (or symbolic) value to an object in order to characterize one attribute of this objet;
- to refer to the application of this method;
- to refer to the result of this application;
- to refer to the process from the design of a measurement method to its exploitation.

For the sake of clarity in this paper, the terms "measure" and "measurement" will not be used alone, but will be used in expressions such as "measurement method", "application of a measurement method", "measurement results" and "measurement process".

Actually, the first three uses of the words "measure" and "measurement" listed above correspond to different steps in a measurement process, and a fourth step, corresponding to the utilization of the result, should be added, as illustrated in Figure 1:
- Step 1: Before measuring, it is necessary to design a measurement method.
- Step 2: The rules of the measurement method are applied to a software or piece of software;
- Step 3: The application of the measurement method rules produce a result.
- Step 4: The measurement result is exploited in a quantitative or qualitative model.

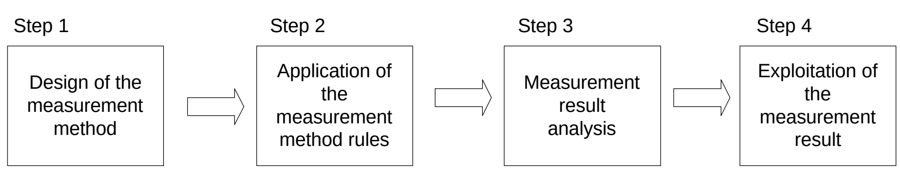

Figure 1: Measurement Process - High-level Model

3. Measurement Process: Detailed Model

This high-level model of a measurement process is the point of view selected as a basis for the design of a validation framework discussed in this paper. An extensive literature review was conducted in order to analyze each step of the measurement process model specified above. This literature review, from within and from outside the software engineering domain, has permitted identification of the required substeps within each of the proposed steps. The set of substeps identified is described next and is illustrated in the detailed model of the measurement process described in Figure 2.

3.1 Step 1 : Design of the measurement method

Four substeps have been identified as being required for a complete design of a measurement method: the definition of the objectives, the design or selection of a metamodel for the object to be measured, the characterization of the concept to be measured and the definition of the numerical assignment rules. The modeling of these substeps, as illustrated on the left-hand side of Figure 2 is based on a logical perspective and on the recognition that iterative and successive refinements are required to improve the initial design proposals of measurement methods. It is also based on the assumption that the measurement method should be built starting with the definition of objectives and ending with the numerical characterization of a software attribute. The four substeps are presented next.

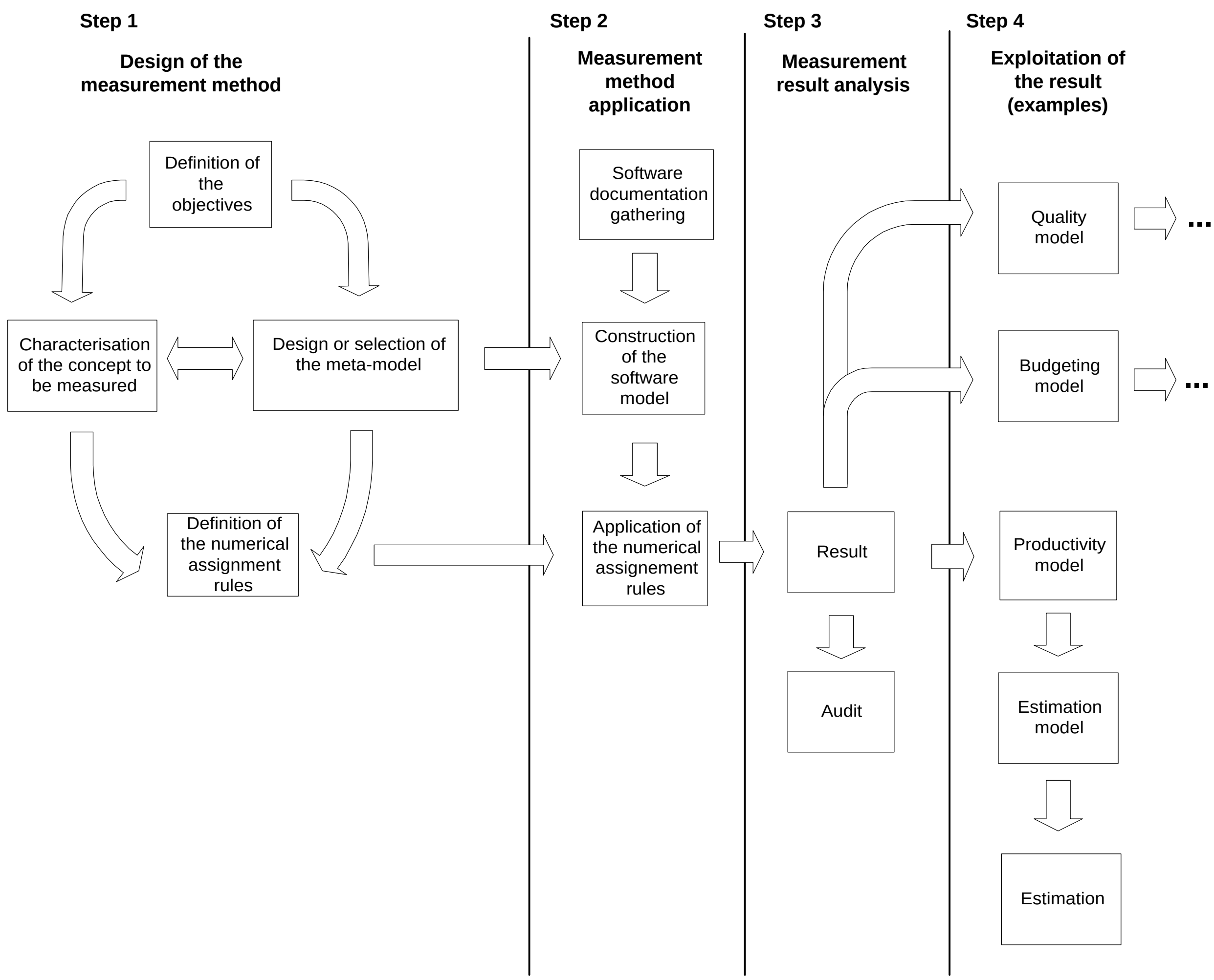

Figure 2: Measurement Process - Detailed Model

• ***Substep 1:*** Definition of the objectives.

Before designing a measurement method, it is important, among other things, to know what we want to measure (what kind of software, which attribute, etc.), what the measurement method point of view will be (software user, software designer, etc.), and what the intended uses are of the measurement method. All of these criteria have a strong influence on the design of the measurement method and consequently will play a non negligible role in the validation process.

• ***Substep 2 :*** Design or selection of the metamodel.
Software is not a tangible product. However, it can be made visible through multiple representations (e.g. for a user, a set of reports, screens etc.; for a programmer, a set of LOC, etc.). The set of characteristics selected to represent a software or a software piece, and the set of their relationships, constitute the metamodel proposed for the description of the software to which the proposed measurement method will be applied. The metamodel must therefore describe the entity types that will be used to describe the software and the rules that allow the identification of the entity types. For example, in the Function Point measurement method (FPMM), an ILF is a piece of software of the entity type. This entity type is defined according to the FPMM`s metamodel. The IFPUG manual presents a definition of this ILF entity type, as well as identification rules to ensure that each and every ILF can be clearly recognized within a software.

• ***Substep 3 :*** Characterization of the concept to be measured.
In order to enable the measurement method to be built, the concept to be measured must be clearly defined. For example, if we consider two distances, we are able to say whether theses distances are equal or not, and, if not, which is bigger. We are able to do this without using numbers, i.e. without measuring, because the concept of the distance between two points is clear and precise.
The definition of a concept can be made in several ways, depending on the nature of this concept. When measuring infinite sets of objects one generally defines some rules which allow characterization of the concept. For some attributes, like the distance between two points, these rules are quite simple. For others, such as abstract attributes - for example, quality or functionality - it may be more complicated. In these cases, the definition can be made by stating explicitly how the concept is decomposed into subconcepts [7]. This decomposition describes which role each subconcept plays in the constitution of the concept measured and how these subconcepts are themselves defined.

From another viewpoint, the mathematical one, to define a concept is to define an empirical relational set [8, 16]. To complete the design of a measurement method, a numerical relational set and an homomorphism between these two relational sets must be defined. This will be carried out in the next sub-step.

A note on substeps 2 and substeps 3 :
It is important to note that the two substeps “definition of the concept to be measured” and “design of the metamodel” are strongly related. Indeed, the definition of a concept cannot be achieved without a representation of the type of objects to be measured, i.e. without a metamodel. On the other hand, the design of metamodels is itself partly dictated by the way the concept will be characterized. Because of the relationships between these two substeps, it was decided that they would be represented here on the same level in our

detailed process model (Figure 2). Although different, one cannot be carried out effectively without the other.

• ***Substep 4 :*** Definition of the numerical assignment rules.
The basis for the numerical assignment rules are the characterization of the concept and of the proposed metamodel. A numerical assignment rule can be described through a descriptive text (a practitioner's description) or through mathematical expressions (a formal theoretical viewpoint). The first type of description will be used when the measurement method is applied. The second will be required in order to allow a mathematical analysis of the mathematical properties of the measurement method. This analysis, carried out by establishing the relationships between the characterization of the concept and the mathematical description of the measurement method will, among other things, enable determination as to whether or not the measurement method is consistently built (internal consistency) and which mathematical operations can be used on the results. Using mathematical expressions, this mathematical analysis will be carried out by proving that an homomorphism exists between the empirical relational set designed in the "definition of the concept to be measured" substep and the numerical relational set implied by the mathematical description of the numerical assignments rules.

The different design substeps defined in Fig. 2 should have been completed by the end of every design process of a measurement method, and all their deliverables should be available and documented, that is: objectives of the measurement method, characterized concept and its decomposition, the metamodel selected for the objects to be measured and the numerical assignment rules.

This logical and successive list of required substeps for the design of a measurement method, as evidenced in other fields of science, is in marked contrast to current practices with various software metrics proposals, where too often the designer is limited to the definition of some numerical assignment rules and then attempts to Analyse whether or not the resulting numbers bear some relationship to other attributes.

3.2 Step 2 : Measurement Method Application

Once the measurement method has been designed, and all its design deliverables are available, it can be applied. The application of a measurement method is carried out through the following substeps (Figure 2):
- Software documentation gathering;
- Construction of the software model;
- Assignment of the numerical rules.

Each of these three substeps in the application of a measurement method is described below.

• *Substep 1 :* Software documentation gathering

The documentation required for the application of the measurement method is collected from the software to be measured. This documentation gathering process will mainly allow the second substep to be carried out, the modeling of the software if the appropriate model of this software is not readily available.

• ***Substep 2 :*** Construction of the software model.
Once this documentation has been gathered, the software model is built. This model describes how the software to be measured is represented by the measurement method. The basis for the construction of the model is, of course, the proposed metamodel, and the rules to model it are the rules identifying the valid components that will take part in the measurement. If the appropriate model has already been built and is available from the previous substeps, this second substep is bypassed.

• ***Substep 3 :*** The numerical assignment rules are next applied to the software model derived from the previous substeps.

3.3 Step 3 : Result of the measurement application

The application of the numerical assignment rules enables a measurement result to be obtained. This measurement result must then be documented and audited. This step can therefore be decomposed into two substeps, as described below.

• ***Substep 1 :*** Presentation of the measurement results.
Applying the measurement rules makes it possible to obtain a result. In order to be evaluated, this result should generally be documented (unit, description of the sub-results, description of the measurement process and team, etc.).

• ***Substep 2 :*** Audit of the results
The results should now be audited according to different methods to ascertain their quality. For example, the tricky parts of mathematical calculations should be checked. The results can also be compared to other well-known results in order to try to evaluate their correctness [14].

3.4 Step 4 : Exploitation of the result

The result of the application of the measurement method can be used in many different ways, many of which might not have been foreseen at the design stage. The right-hand column of Figure 2 illustrates a few potential uses of a software measurement result: in quality models, in budgeting models, or in an estimation process which is itself based on a productivity model and an estimation model. This substep is much better known and will not be discussed further in this paper.

4. Measurement Process: Validation Requirements

The previous section has presented a detailed process model for software measurement methods. From this model, a validation framework could be designed, as well as a validation guide, for analyzing whether or not a software metrics could qualify as a measurement method. The process model presented could also be used for analyzing the coverage of previous validation methods proposed for software metrics. These topics are discussed next.

4.1 Validation Framework

A validation framework of software metrics from a measurement method point of view should take into account all the steps of the measurement process model presented in the preceding sections and illustrated in both Figure 1 and Figure 2. This does not mean that the validation of the design of a measurement method has to be carried out every time a measurement method is used, but that the validation step must be applied at least once. A good evaluation of a measurement method would require, in most cases, a good knowledge of the ways in which the different steps and substeps described in Figure 2 have been achieved. It should be possible, for a given measurement method and in a validation perspective, to provide information about each of these steps and substeps, and, where required, documented evidence of whether or not the specified criterion has been met. Three examples are presented below.

- Example 1: The intended use of the measurement method can impact the precision level to be required from this method. Thus, a particular degree of precision may be adequate for one measurement method, but inadequate for another, according to their intended uses. In order to be able to evaluate the adequacy of the degree of precision of a method, the objectives of this method must be known (step 1, substep “objectives of the method ”).

- Example 2: The internal validity of a measurement method will be demonstrated mainly while establishing links between the “definition of the concept” and the “definition of the numerical assignment rules” substeps (see above). Internal validation of a measurement method must then take into account information about these two steps.

- Example 3: It is said that a measurement method is repeatable if it provides the same result when applied on the same software by people with the <u>same</u> skillset. The repeatability of a measurement method may be tested in a practical way by applying the method to one or several software, and the result will depend on the way the metamodel and the numerical assignment rules are designed. Once again, the decision on whether or not a measurement method is repeatable is made through an analysis of the “characterization of the concept”, the “numerical assignment rules” and the “application of the measurement method” steps and substeps.

4.2 Validation Guide

A guide providing the list of check-points for all operations of each step and substep described in the detailed model of a measurement process would be a non negligible value in elaborating a comprehensive validation framework for a measurement process .

For example, for the substep “design of the metamodel”, some of the points to be checked might be :

- The proposed description of the metamodel is consistent with the software to be measured;
- The metamodel states the relationships, if any, between entity types;
- The definition of the entity types of the metamodels are defined without ambiguities;
- The identification rules for the entity types are not ambiguous;
- etc.

In order to become a complete validation guide, this guide should also :
- explain how these check-points can be verified;
- establish links between the validation questions (Is the measurement repeatable? What is the error rate of the measurement method?, etc.) and the various steps of the cycle. This could be achieved by describing, for each step and substep, what must be checked in order to answer these questions. Work is progressing on the design of such a guide, and its first version will be field tested in 1997.

4.3 Analysis of Metrics Validation Proposals

As pointed out in the introduction, many authors have addressed the validation problem from different points of view, but a complete framework addressing the different parts of the measurement method process as described in Figure 2 is still missing. For example, Schneidewind [17] seems to propose a validation process based mainly on the analysis of the results of measurement methods (Step 3 of Figure 2: Measurement Result Analysis). Fenton [8], for his part, suggests that, in order to be valid, a measurement method must satisfy the representation condition of measurement theory. In other words, he proposes a validation process which addresses only the relationships between the two substeps “characterization of the concept to be measured” and “definition of the numerical assignment rules” of the first step in Figure 2. On the other hand, Kitchenham et al. [11] propose a validation process addressing some parts of the first three steps in Figure 2. Nevertheless, this validation framework does not cover the full spectrum of the process model of measurement methods proposed here. For example, it does not tackle the validation of the metamodel and its relationships with the different components of measurements methods.

5. Conclusion

Unfortunately, it looks like many of the software metrics proposed have not gone through such a design cycle and through intensive, adequate and relevant data verification and audit, at either the high-level or at the detailed level of software measurement method validation.

Indeed, many software metrics have been based on a unverified “intuitive” approach and are not based upon verifiable foundations. Therefore, a significant number of so-called software metrics would not qualify as measurement methods. For example, many software metrics do not formally specify their metamodel, using terms like “process”, “flows of data” without providing precise definitions for them. This lack of exactness explains that the results given by these measures are context-dependent, i.e. dependent on the way the users of the measures understand the terms.

Furthermore, it appears that even for far more sophisticated measurement methods, such as Function Point Analysis (FPA), the design deliverables are not yet all available. For this FPA, even though the metamodel and the numerical assignment rules are properly defined, it seems that the “characterization of the concept to be measured” step has not been totally clarified. Consequently, the mathematical links between this concept and the numerical assignment rules are not yet fully established (this does not mean, of course, that they cannot be established). This implies that no one has yet been able to prove, in a formal way, that this measurement method is consistent and that it measures what it is supposed to measure. Notwithstanding this fact, users of the method have provided a significant amount of evidence on the effectiveness of the FPMM with prediction systems. This would mean that while the internal validity of FPMM has not been proved, there is some evidence that external validity has been demonstrated.

Designing a complete validation framework for measurement methods is a highly complex challenge. This article has described the steps and sub-steps of a measurement process model proposed as the basis on which a validation framework of a “measurement method” should be based.

References

[1] A. Abran.
Analyse du processus de mesure des points de fonction. Thèse de doctorat de l’université de Montréal, 1994.

[2] A. Abran, E. Ahki.
Validation requirement for functional size measurement. Internal Report, Research Laboratory in Software Engineering, Université du Québec à Montréal, 1994.

[3] A. Abran, P. Robillard.
Function Points: a Study of Their Measurement Processes and Scale Transformations, Journal of Systems and Software, vol. 25, issue 2, p. 171, 1994.

[4] J.M. Bieman, J. Schultz.
« An Empirical Evaluation (and Specification) of the all-du-paths. Testing Criterion. Software Engineering Journal, Vol. 7. No. 1, pp. 43-51, January 1992.

[5] L. Briand, K. El Emam, S. Morasca.
“On the Application of Measurement Theory in Software Engineering”, International Software Engineering Research Network, Technical report #ISERN-95-04.

[6] C.H. Coombs, R.M. Dawes, A.Tversky.
"Psychologie mathématique", Tome 1, Presses Universitaires de France (PUF), "La Psychologie d'aujourd'hui", Paris 1975.

[7] P. Dickes, J. Tournois, A. Fieller, J-L. Kop.
"La psychométrie", Presses Universitaires de France (PUF), collection "Le Psychologue", Paris 1994.

[8] N. Fenton
"Software Metrics : A Rigorous Approach", Chapman & Hall, 1991.

[9] N. Fenton and B. Kitchenham, "Validating Software Measures", J. of Software Technology, Verification and Reliability, vol 1, no. 2, pp. 27-42, 1991.

[10] S.H. Kan
"Metrics and Models in Software Quality Engineering", Addilson-Wesley Publishing Compagny, 1994.

[11] B. Kictchenham, S. L. Pfleeger, N. Fenton.
"Towards a Framework for Software Measurement Validation", IEEE Transactions On Software Engineering, vol 21, dec. 1995.

[12] B.S. Massey
"Measures in Sciences and Engineering, their Expression , Relation and Interpretation". Ellis Horwood Series in Mathematics and its applications. Ellis Horwood Limited.

[13] M. Maya.
"Mesure de la taille fonctionnelle des logiciels en temps réel. Revue critique de la littérature Internal Report, Research Laboratory in Software Engineering, Université du Québec à Montréal, août 1996.

[14] P.M. Morris, J.M. Desharnais
"Function point analysis. Validating the results", IFPUG Spring Conference, Atlanta, April 1996.

[15] Pfanzgal J.
"Theory of Measurement", Wiley, New York, 1968.

[16] Roberts F.S.
"Measurement Theory", Encyclopedia of Mathematics and its Applications, Vol. 7, section Mathematics and the Social Sciences, Addison-Wesley Publishing Compagny, Massachusetts.

[17] Roberts F.S., C.H.
"On the Theory of Uniqueness in Measurement ", Journal of Mathematical Psychology, 14, 1976, p. 211-218.

[18] N. Schneidewind
"Methodology for Validating Software Metrics", IEEE Transactions on Software Engineering, Vol. 18, no. 5, pp. 410-442, May 92.

[19] M. Shepperd, Darrel Ince
"Derivation and Validation of Software Metrics", Oxford Science Publications, 1993.

[20] P. Suppes, J.L. Zinnes
"Basic Measurement Theory", Handbook of Mathematical Psychology, Edited by D.D. Luce, R.R. Bush E. Galanter, Volume I, Wiley, New York 1963, p. 1-76.

[21] Horst Zuse.
Foundations of Validation, Prediction and Software Measures, Proceedings of the AOSW94 (Annual Oregon Software Metric Workshop, Portland, April 20-22, 1994.

[22] "Definition of Size Measurement". Document ISO/IEC DIS 14143-1

[23] Document ISO ISO/IEC Guide 25
"General Requirements for the Competence of Calibration and Testing Laboratories",
International Standards Organization , International Electrotechnical Commission, 1990.

[24] "Function Point Counting Practices Manual", International Function Point Users Group (IFPUG), Version 4.0, 1994.