A Roadmap to Maturity for Software Measures

Alain Abran

École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest, Montréal, Canada H3C 1K3
alain.abran@etsmtl.ca

Abstract. Up until recently ‘software metrics' have been most often proposed as the quantitative tools of choice in software engineering, and the analysis of these had been most often discussed from the perspective referred to as ‘measurement theory’. However, in other disciplines, it is the domain of knowledge referred to as ‘metrology’ that is the foundation for the development and use of measurement instruments. This paper presents an overview of the set of metrology concepts as documented in the ISO Vocabulary of Basic and General Terms in Metrology (VIM) and its use in analyzing ‘software metrics’. It also presents the measurement coverage within the Guide to the Software Engineering Body of Knowledge (SWEBOK) as well as a proposed measurement body of knowledge. Throughout these analyses some gaps are identified which need to be addressed for software measurement to mature.

Keywords: Software Measures, Software Metrics, Metrology, SWEBOK, Measurement Body of Knowledge.

1 Introduction

In the field of software engineering, the term “metrics” is used in reference to multiple concepts, whether in terms of the quantity to be measured (measurand[1]), measurement procedures, measurement results or models of relationships across multiple measures, or of the objects themselves. In the software engineering literature, the term is applied, for instance, to a measure of a concept (e.g. McCabe cyclomatic complexity), to quality models (ISO 9126 – software product quality) and to estimation models (e.g. Halstead’s effort equation, COCOMO estimation models). This has led to many curious problems, among them a proliferation of numerous publications on metrics for concepts of interest, but with a very low rate of acceptance and use by either researchers or practitioners, as well as a lack of consensus on how to validate so many proposals. The inventory of software metrics is at the present time so diversified and includes so many individual proposals that it is not seen to be economically feasible for either industry or the research community to investigate each of the hundreds of alternatives proposed to date.

[1] A measurand is defined as a particular quantity subject to measurement; the specification of a measurand may require statements about quantities such as time, temperature and pressure [ISO VIM].

In software engineering, the 'software metrics' approach has been up until fairly recently the dominant approach to software measurement. Most of these ‘metrics’ have been designed based either on the intuition of the researchers or on an empirical basis, or both. In their analysis of some of these ‘metrics’, researchers have most often used the concepts of 'measurement theory' as the foundation for their analytical investigation. However, while relevant, 'measurement theory' deals with only a subset of the classical set of concepts of measurement; 'software metrics' researchers, by focusing solely on measurement theory, have investigated mainly the representation conditions, the mathematical properties of the manipulation of numbers and the proper conditions for such manipulations [FENT97, ZUSE97].

Our survey of the literature on ‘software metrics’ has come up with almost no references to the classical concepts of metrology in these investigations into the quality of the ‘metrics’ proposed to the software engineering community. Only recently has some of the metrology related concepts been introduced in the ISO software engineering standards community. Is software measurement itself a mature tool set and can metrology help to investigate this research topic?

This paper presents in Section 2 the modeling of the sets of concepts in the ISO vocabulary of terms in metrology (VIM) and in section 3, examples of our use of the VIM in the analysis of ‘software metrics’. Section 4 presents an analysis of the coverage of measurement within SWEBOK and section 5, our proposed measurement body of knowledge to address some of the measurement gaps identified in the SWEBOK Guide. A discussion is presented in section 6.

2 Metrology

2.1 ISO Metrology concepts

In engineering as well as in other fields such as business administration and a significant number of the social sciences, measurement is one of a number of analytical tools. Measurement in these other sciences is based on a large body of knowledge; such a body of knowledge, built up over centuries and millennia, is commonly referred to as the field of 'metrology'. This domain is supported by government metrology agencies, which are to be found in most industrially advanced countries.

The ISO document that represents the official international and legal consensus is the ISO Vocabulary of Basic and General Terms used in Metrology [ISO VIM]. While this key ISO document is widely known in the field of metrology, it is almost unknown in the 'software metrics' community. This ISO VIM follows some of the concepts of the traditional presentation of vocabularies, with 120 terms described individually in textual descriptions. However, this mode of representation is challenging in terms of assembling the full set of interrelated terms; to improve the presentation and the understanding of this complex set of interrelated concepts, we presented in [ABRA02a] an initial set of models for the various levels of metrology concepts within the ISO Vocabulary.

The high-level model of the set of categories of terms is presented in Figure 1. This model, together with some sub-models presented later on, corresponds to our current understanding of the topology integrated into the vocabulary of this specialized area of the body of knowledge relating to metrology. To represent the relationships across the terms, the classical representation of a production process was selected: e.g. input, output and control variables, as well as the process itself inside the box. In Figure 1, the output is represented by the 'measurement results' and the process itself by the 'measurement' in the sense of measurement operations, while the control variables are the 'étalons'[2] (official yardsticks) and the 'quantities and units'. This set of concepts represents the 'measuring instrument'. It is to be noted that the measurement operations, and, of course, the measurement results, are influenced by the 'characteristics' of the measuring instruments. In the VIM, the term 'measurements' used as a single term corresponds to the 'set of operations' used for measuring.

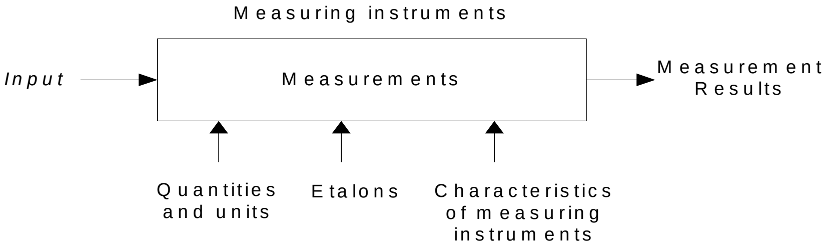

Figure 1: Model of the categories of metrology terms [ABRA02a]

The term 'metrology' includes all aspects of measurement (theoretical and practical), collectively referred to in the metrology literature as the science of measurement (Figure 2). Metrology encompasses the 'principles of measurement', which represent the scientific basis for measurement. From the principles of measurement, the 'method of measurement' in the general sense is then instantiated by a measurement as a set of operations. Figure 2 depicts this hierarchy of concepts.

The detailed topology of the measurement process is instantiated next in a 'measurement procedure' (Figure 3), again as a process model having the 'measurand' as its inputs, control variables and an output representing the 'measurement results'. To carry out a measurement exercise, an operator should design and follow a 'measurement procedure' which consists of a set of operations, specifically described, for the performance of a particular measurement according to a given measurement method. The instantiation of a measurement procedure handles a 'measurement signal' and produces a transformed value, which represents a given measurand. The results of the measurement can be influenced by an 'influence quantity' during the measurement

[2] Étalon: for instance, an internationally recognized material yardstick: the physical 'meter' etalon in length measurement recognized as the official 'étalon' for the meter. Étalons are also refined over time: for instance, the official definition of the meter has changed in 1983: it was then defined as the distance performed by the light, in an empty medium, in 1/299 792 458 second.

process: for example, the temperature of a micrometer during the measurement of the length of a particular object.

The category 'measurement results' is presented next in the form of a structured table according to the types of measurement results, the modes of verification of the measurement results and information about the uncertainty of measurement – Table 1.

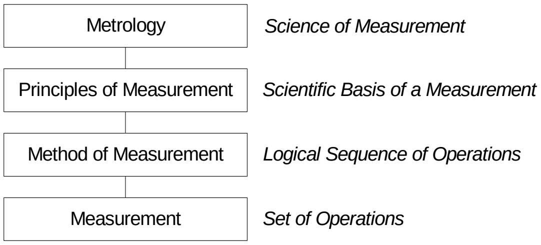
Metrology
Science of Measurement
Principles of Measurement
Scientific Basis of a Measurement
Method of Measurement
Logical Sequence of Operations
Measurement
Set of Operations

Figure 2: Measurement foundations [ABRA02a]

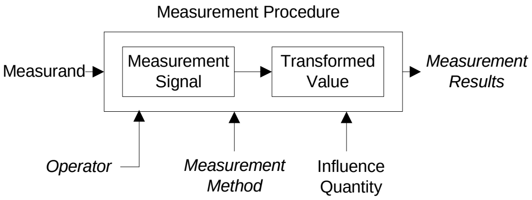

Figure 3: Measurement Procedure [ABRA02a]

Table 1: Classification of terms in the category of 'Measurement Results' [ABRA02a]

Types of measurement results	***Modes of verification of measurement results***	**Uncertainty of measurement**	
Indication (of a measuring instrument) Uncorrected result Corrected result	Accuracy of measurement Repeatability (of results of measurements) Reproducibility (of results of measurements)	Experimental standard deviation Error (of measurement) Deviation	Relative error Random error Systematic error Correction Correction factor

3 Metrology as a tool to analyze 'software metrics'

3.1 Related work

While metrology has a long tradition of use in physics and chemistry, it is rarely referred to in the software engineering literature. Carnahan et al. [CARN97] are among the first authors to identify this gap in what they referred to as "IT metrology"; they highlight the challenges and opportunities arising from the application of the metrology concepts to information technology. In addition, they have proposed logical relationships between metrology concepts, consisting of four steps to follow to obtain measured values: defining quantity/attribute, identifying units and scales, determining the primary references and settling the secondary references. Moreover, Gray [GRAY99] discusses the applicability of metrology to information technology from the software measurement point of view.

[ABRA02b, 03] has highlighted some high-level ambiguities in the domain of software measurement, and proposed substituting the appropriate metrology terms for the current ambiguous and peculiar software metrics terminology unique to the domain of software engineering. In metrology, the term "metrics" is never used. Moreover, Sellami and Abran [SELA03] have investigated the contribution of metrology concepts to understanding and clarifying the framework for software measurement validation proposed by Kitchenham et al. in [KITC95].

3.2 Functional Size measurement

In the ISO software engineering community, there has been work to investigate and apply the metrology concepts to software measurement standards, including within specific measurement methods. The first type of measurement methods tackled at the ISO level were the functional size measurement methods with the publication of ISO meta-standards on functional size measurement [ISO 14143-1], dealing with some of the design issues of measurement method:

1. Part 1 of 14143: dealing with the ISO definitions of concepts for functional size measurement.
2. Part 2 of 14143: dealing with conformity assessment of the design of proposed functional size methods.
3. Part 3 of ISO 14143: dealing with the verification criteria of a functional size method to assist measurements users in selecting the methods most appropriate to their needs.
4. Part 4 of ISO 14143: providing a large set of functional user requirements against which candidate measurement methods can be tested.
5. Part 5 of ISO 14143: providing users with the information for analyzing which measurement method is most appropriate to the functional domain of the software to be measured.
6. Part 6 of ISO 14143: providing users with the information for selecting a specific measurement method according to their needs.

In addition, four specific methods have been recognized as ISO international standards, that is: IFPUG [ISO 20926], NESMA, MKII and COSMIC [ISO 19761], a second generation functional size measurement method. Many of the metrology related concepts have already been integrated into the design of the COSMIC method (ISO 19761), with particular attention paid to the characterization of the concept being measured, to the selected meta-model of the functionality, and to the units and quantities in the definition of the numerical assignment rules. Sellami and Abran [SELA05, ABRA97] have documented the metrology concepts addressed in ISO 19761 (COSMIC-FFP), both in the design of this measurement method and in some of its practical uses.

3.3 ISO 9126 'quality metrics'

The evaluation approach of ISO TR 9126-4 in [ABRA06] was based on an evaluation process previously used for investigating the design of cyclomatic complexity [ABRA04a], Halstead's metrics [ALQU05], IFPUG function points [ABR94, 96] and Use Case Points [OUW06].

The ISO 9126 series of documents on software product quality evaluation proposes a set of 120 metrics[3] for measuring the various characteristics and subcharacteristics of software quality. However, as is typical in the software engineering literature, the set of so-called metrics in ISO 9126 refers to multiple distinct concepts. To help in understanding and clarifying the nature of the metrics proposed in ISO TR 9126-4, each was analyzed from a metrology perspective and mapped to the relevant metrology concepts. Such an evaluation approach also contributes to identifying the measurement concepts that have not yet been tackled in the ISO 9126 series of documents. Each of these gaps represents an opportunity for improvement in the design and documentation of the measures proposed in ISO 9126. Based on the analysis using metrology concepts in [ABRA06] the following comments and suggestions were made:

- Identifying and classifying the 'quality in use metrics' into base and derived quantities makes it easy to determine which should be collected (base quantities) to be used subsequently in computing the other (derived) quantities.
- Some of the ISO 9126-4 derived units depend on other quantities with unknown units and are therefore ambiguous.
- None of the 'quality in use metrics' refers to any system of units, coherent (derived) unit, coherent system of units, international system of units (SI), off-system units, multiple of a unit, submultiple of a unit, true values, conventional true values or numerical values.
- None of the base and derived quantities, except for task time, has symbols for their measurement units.

[3] While the term "metrics" is used in ISO 9126, the use of this term will be abandoned and replaced by "measures" in the next ISO version currently in preparation as an initial step towards harmonizing the software engineering measurement terminology with the metrology terminology.

4 Measurement within SWEBOK

4.1 Measurement in SWEBOK

Measurement is embedded within the IEEE Computer Society definition of software engineering as

“(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
(2) The study of approaches as in (1)” [IEEE 610].

The topic of measurement within SWEBOK was one of the editorial criteria for the initial write-up. The SWEBOK Knowledge Area editors were expected to adopt the position that the measurement 'theme' is common across all Knowledge Areas, and therefore had to be incorporated into the proposed breakdown of topics in each Knowledge Area. Since the acceptance criterion for inclusion in Guide to the SWEBOK was 'generally accepted', it is important to ask what did in fact gain an approval on a consensual basis with respect to measurement, and what can be learned from this consensus, or the lack of it. It is worth reminding that the 'generally accepted' definition adopted in SWEBOK originates from the Project Management Institute (PMI), that is: 'applies to most of the projects, most of the time, and widespread consensus validates its value and effectiveness'.

4.2 A measurement process model

In their work as ISO editors for the Guide to the Verification of Functional Size Measurement Methods [ISO 14143-3], Abran and Jacquet studied the various software engineering authors dealing with 'metrics validation' [JACQ97, 99]. Significant variations were found in the authors' approaches as well as the use of similar terms by these authors, but with very significant differences in the related concepts.

To clarify the confusion due to the inconsistent terminology used by these authors, a broader measurement process model was proposed (Figure 4) identifying 4 distinct steps, from the design of a measurement method to the exploitation of the measurement results [JACQ97, 99]. Then, the approaches of the various authors, as well as the validation concepts that were being addressed differently by these authors, were sorted out depending on whether or not they were addressing validation issues related to Steps 1 to 4 of the process model in Figure 4.

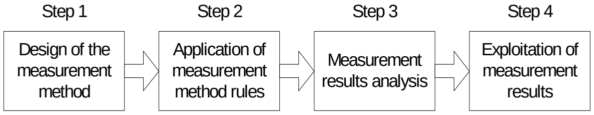
Step 1
Design of the measurement method
Step 2
Application of measurement method rules
Step 3
Measurement results analysis
Step 4
Exploitation of measurement results

Figure 4: Measurement Process – High-level Model

It is to be noted that very few of the measurement concepts present in the ISO VIM address the first step (design of a measurement method) and none address the last step (exploitation of the measurements results) of Figure 4. This is illustrated in Table 2, which depicts a partial mapping between Figures 1 and 4: for instance, for the design of a measurement method, the Abran and Jacquet model includes more concepts than simply 'quantities and units'.

Table 2: Alignment of metrology concepts with the measurement process model

Measurement process model [JACQ 97, 99]	**Step 1** Design of Measurement Methods	**Step 2** Application of measurement method rules	**Step 3** Measurement results analysis	**Step 4** Exploitation of measurement results
ISO metrology model [ABRA02]	• Quantities and units	• Measuring instruments • Characteristics of measuring instruments	• Measurement results	

4.3 Metrology concepts in SWEBOK

Using both the ISO set of metrology concepts model and the measurement process model [JACQ97, 99], the current status of the field of software measurements as documented in the SWEBOK Guide was analyzed [ABRA04b]. The results of the analysis of the presence of metrology concepts within each KA are presented in Table 3. Using a detailed inventory of the measurement-related statements appearing in the ten SWEBOK chapters, these statements were analyzed in terms of measurement concepts, and then mapped into both the set of metrology concepts presented in Section 2 and to the measurement process model presented in 4.2.

Table 3 lists, for each of the ten chapters of SWEBOK, which metrology concepts and measurement steps are addressed whenever a measurement-related statement appears in the SWEBOK Guide. From Table 3, it can be observed that a large majority of the measurement-related concepts mentioned in SWEBOK are listed in the category of concepts related to the exploitation of the measurement results. Very few SWEBOK statements directly address the measuring instrument or the quality of the direct measurement results (prior to their use in quantitative analytical models (assessment models or predictive models)). And only one measurement related statement in the Software Quality chapter addresses a single aspect of the design of measurement instrument, and only through a subset of the metrology concepts of quantities and units.

Table 3: Measurement steps and metrology category of concepts within SWEBOK [ABRA04b]

SWEBOK **Knowledge Area**	**Measurement Steps** **Topics**	**Step 1** Design of measurement methods (Quantities and units)	**Step 2** Application of measurement method rules (Measuring instruments)	**Step 3** Measurement results analysis	**Step 4** Exploitation of measurement results
Software engineering requirements	Process support and management				×
	Requirements negotiation				×
	Document quality				×
	Acceptance tests				×
	Requirements tracing				×
Software engineering design	Measures			×	
Software engineering testing	Evaluation of the program under test				×
	Evaluation of the tests performed				×
Software engineering maintenance	Software Maintenance Measurement				×
Software configuration management	Surveillance of software configuration management				×
Software engineering management	Goals				×
	Measurement Selection				×
	Measuring Software and its Development				×
	Collection of data		×		
	Software Measurement Models			×	
Software engineering process	Methodology in process measurement		×		
	Process Measurement Paradigms				×
Software engineering quality	Measuring the value of quality				×
	Fundamentals of Measurement	×			
	Measures			×	
	Measurement analysis techniques				×

	Defect characterization				×
	Additional Uses of SQA and V&V data				×

This highlights the fact that, even though the use of measurement results is quoted in most KA, both the KA editors and the extensive number of reviewers have not been able to come up and agreed on the availability of knowledge on measurement concepts which met the SWEBOK and PMI criteria of generally accepted, that is of 'applies to most of the projects, most of the time, and widespread consensus validates its value and effectiveness'. This does not mean that such other types of measurement knowledge do not exist in the literature, but rather that there is not yet a wide consensus on their value and effectiveness and their generalization power outside of the initial context of operations. It also points out to a significant lack software measurement methods with enough strengths as measurement instruments and meeting the metrology criteria for quality (of measuring instruments). Table 3 also points out to a lack of widely recognized and validated quantitative data to support yet the quality expected from an engineering viewpoint for the software engineering topics described.

This, of course, corresponds to a lack of recognized references to other measurement concepts from the recognized body of knowledge on metrology. This is a clear indication that, when looked at from an engineering perspective, measurement in software engineering is far from being mature and that it currently constitutes a fairly weak engineering foundation for the field of software engineering.

5 Measurement body of knowledge

Measurement is, of course, fundamental to the engineering disciplines, and, at the inception of the SWEBOK, it had been given to all the KA associate editors as a criterion for identifying relevant measurement-related knowledge in their respective Knowledge Areas. Individual associate editors initially developed each of the 10 Knowledge Areas on their own, and even though a large number of reviewers contributed in the numerous reviews, this still led to different levels of breadth and depth of treatment of subtopics like measurement: therefore, measurement-related knowledge has not been developed equally across Knowledge Areas. Subsequently, we proposed a unified view of the measurement knowledge in software engineering [ABRA04b, BUGL05] in the form of a proposal for a distinct KA on Software Measurement, taking into account all the measurement-related items from the 2004 version of the SWEBOK Guide [ABRA05]. This proposal is shown in Fig. 5.

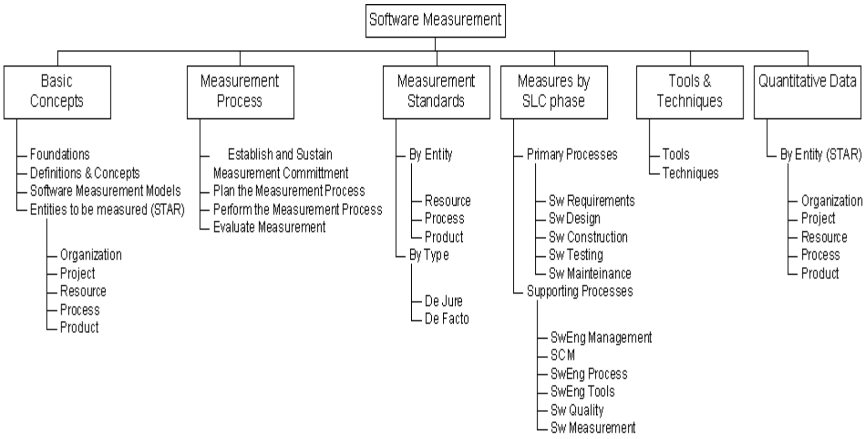

Figure 5 – Proposed breakdown for the Software Measurement KA [ABRA04b, BUGL05]

To investigate the credibility of the recommended reference material for our proposal for an additional KA on Software Measurement, the level of empirical support as documented in the SWEBOK references was investigated next, including twenty four (24) additional ones recommended in [BUGL05] to cover the “gaps” in the measurement references. These references have been grouped in three types:

- **International standards** (ISO, IEEE or other standards organizations): These are based on international consensus by either technical experts or ISO-recognized voting countries, or both.
- **Books**: These often represent only the author’s opinions. A book also contains a number of chapters, each of which could be based on a different type or types of empirical support.
- **Papers and book chapters**[4]: The most relevant empirical support method is mentioned. When there is not a direct mapping to one of the 12 empirical support methods proposed by [ZELK98], the “not applicable” code has been assigned.

To reach a point where this measurement body of knowledge would be recognized as generally accepted in the broader software engineering community, it is mentioned in [BUGL05] that further steps are required to get this measurement taxonomy validated by peers in the software engineering measurement community.

[4] This is an interim classification.

6 Discussion

Currently 'software metrics' are most often proposed as the measurement tools of choice in empirical studies in software engineering. While this field of 'software metrics' has most often been discussed from the perspective referred to as 'measurement theory', in other disciplines, however, it is the domain of knowledge referred to as 'metrology' that is the foundation for the development and use of measurement instruments and measurement processes. Measurement is recognized as a key element of engineering and, because of design criteria in the Guide to SWEBOK, it is pervasive in the Guide. But, is software measurement already a mature tool set?

In this paper, we have identified analytical tools to investigate the state of the art of measurement in software engineering, focusing on the set of metrology concepts. Both our initial modeling of the sets of measurement concepts documented in the ISO International Vocabulary of Basic and General Terms in Metrology and our measurement process model were used to survey, and position, the measurement-related statements in the Guide to the Software Engineering Body of Knowledge. This has revealed that, even though measurement-related statements appear throughout the SWEBOK document, they overwhelmingly concern the use of measurement results in assessment and predictive models. By contrast, there is in this document very little widely recognized validated knowledge from an engineering perspective, little on the quality of the quantitative inputs to these models, and almost nothing on supporting measuring instruments necessary to obtain these inputs.

Similarly, in the software engineering literature, even though there is a large number of 'metrics' proposed, there is still very little discussion on the topic of measuring instruments so overwhelmingly present in the traditional engineering disciplines.

This also illustrates that most of the metrology concepts, and sub-concepts, have not yet been discussed or addressed to a significant extent in the 'software metrics' literature. In the context where measuring instruments are necessary key elements of empirical studies, this points to a potentially significant weakness in current empirical studies in software engineering, while at the same time providing an indication of where metrology-related improvements in software measurement could contribute significantly to strengthening future empirical studies in software engineering.

This analysis from the metrology perspective suggests that the field of software measurement has not yet been fully addressed by current research, and that much work remains to be done to support software engineering as an engineering discipline based on quantitative data and adequate measurement methods meeting the classic set of criteria for measuring instruments as described by the metrology body of knowledge in large use in the engineering disciplines.

References

[ABRA94] Abran, Alain; Robillard, P.N., “Function Points: A Study of their Measurement Processes and Scale transformation,” Journal of Systems and Software, Vol. 65, 1994, pp. 171-184.

[ABRA96] Abran, Alain; Robillard, P.N., “Function Points Analysis: An Empirical Study of Its Measurement Processes,” IEEE Transactions on Software Engineering, Vol. 22, no. 12, December 1996, pp. 895-910.

[ABRA97] A. Abran and J.P. Jacquet, "A Structured Analysis of the New ISO Standard on Functional Size Measurement-Definition of Concepts", Fourth IEEE International Symposium and Forum on Software Engineering Standards, 1999, pp. 230-241.

[ABRA98] Abran, A., "Software Metrics Need to Mature into Software Metrology (Recommendations)," NIST Workshop on Advancing Measurements and Testing for Information Technology (IT), Maryland, USA, 1998.

[ABRA02a] A. Abran and A. Sellami, "Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology", in Software Measurement and Estimation, 12th International Workshop on Software Measurement - IWSM, Magdeburg (Germany) Oct. 7-9 2002, Shaker-Verlag, Aachen, 2002, ISBN 3-8322-0765-1, pp. 315.

[ABRA02b] A. Abran and A. Sellami, "Measurement and Metrology Requirements for Empirical Studies in Software Engineering", IEEE Software Technology and Engineering Practice Workshop (STEP), Montreal (Canada), 2002.

[ABRA03] Abran, A., Sellami, A., and Suryn, W., "Metrology, Measurement and Metrics in Software Engineering," 9th International Software Metrics Symposium, Sydney, Australia, 2003.

[ABR04a] Abran, A., Lopez, M., Habra, N., “An Analysis of the McCabe Cyclomatic Complexity Number,” 14th International Symposium on Software Measurement IWSM-Metrikon, Shaker Verlag, 2004, Magdeburg, Germany, pp. 391-405.

[ABRA04b] Abran, A.,Buglione, L, Sellami, A. (2004), “Software Measurement Body of Knowledge – Initial Validation using Vincenti’s Classification of Engineering Knowledge types”,14th International Workshop on Software Measurement IWSM-MetriKon 2004, Konigs Wusterhausen, Germany, Éditeur : Shaker-Verlag, pp. 255-270.

[ABRA05] A. Abran, J. Moore, P. Bourque, R. Dupuis, and L. Tripp, *Guide to the Software Engineering Body of Knowledge - SWEBOK - Version 2004*, IEEE-Computer Society Press: www.swebok.org, Los Alamitos (USA), 2005.

[ABRA06] Abran, A., Al Qutaish, R., Cuadrado, J., “Analysis of the ISO 9126 on Software Product Quality Evaluation form the Metrology and ISO 19539 Perspectives”, WSEAS Transactions on Computers, November Issue, Vol. 5, No. 11, 2006, World Scientific & Engineering Academy and Society (WSEAS), Greece, 2006, pp. 2778-2786. (ISSN: 1109-2750).

[ALQU05] Al Qutaish, R. E., Abran, A., “An Analysis of the Design and Definitions of Halstead’s Metrics,” 15th International Symposium on Software Measurement (IWSM), Shaker Verlag, 2005, Montréal, Canada.

[BUGL05] Buglione, L., Abran, A., “ Software Measurement Body of Knowledge – Overview of Empirical Support”, 15th International Workshop on Software Measurement – IWSM 2005, Montréal (Canada), Shaker Verlag, Sept. 12-14, 2005, pp. 353-368.

[CARN97] Carnahan, L., Carver, G., Gray, M., Hogan, M., Hopp, T., Horlick, J., Lyon, G., and Messina, E., "Metrology for Information Technology," *StandardView*, Vol. 5, No. 3, 1997, pp. 103-109.

[FENT97] N.E. Fenton and S.L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, second edition, PWS Publishing Co., 1997, pp. 638.

[GRAY99] Gray, M. M., "Applicability of Metrology to Information Technology," *Journal of Research of the National Institute of Standards and Technology*, vol. 104, No. 6, 1999, pp. 567-578.

[IEEE 610] IEEE IEEE 610.12: 1990, *IEEE Standard Glossary for Software Engineering Terminology*, IEEE Computer Society, USA, 1990.

[ISO VIM] ISO, *International Vocabulary of Basic and General Terms in Metrology*, International Organization for Standardization - ISO, Geneva, 1993, pp. 49.

[ISO 9126-4] ISO/IEC, *ISO/IEC TR 9126-4: Software Engineering - Product Quality - Part 4: Quality in Use Metrics*, International Organization for Standardization, Geneva, Switzerland, 2004.

[ISO 14143-1] ISO ISO/IEC 14143-1: 1998, *Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*, International Organization for Standardization - ISO, Geneva, 1998.

[ISO 14143-3] ISO ISO/IEC 14143-3: 2003, *Software Engineering- Functional Size Measurement- Part 3: Verification of Functional Size Measurement Methods*, International Organization for Standardization - ISO, Geneva, 2003.

[ISO 15939] ISO ISO/IEC 15939: 2002, *Information Technology - Software Engineering - Software Measurement Process*, International Organization for Standardization - ISO, Geneva, 2002.

[ISO 19761] ISO ISO/IEC 19761: 2003, *Software Engineering -- COSMIC-FFP -- A Functional Size Measurement Method*, International Organization for Standardization - ISO, Geneva, 2003.

[ISO 20926] ISO/IEC 20926: 2003, Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual, International Organization for Standardisation – ISO, Geneva, 2003.

[JACQ97] Jacquet, JP., Abran, A., "From Software Metrics to Software Measurement Methods: A Process Model", *International Software Engineering Standards Symposium and Forum, ISESS 97*: IEEE-Computer Society Press, 1997, pp. 128-135.

[JACQ99] Jacquet, JP., Abran, A., "Metric Validation Proposals : A Structured Analysis", in *Software Measurement - Current Trends in Research and Practice, 8th International Workshop on Software Measurement - IWSM 98*: Deutscher Universität Verlag, 1999, pp. 43-59.

[KITC95] Kitchenham, B., Pfleeger, S. L., and Fenton, N., "Towards a Framework for Software Measurement Validation," *IEEE Transaction on Software Engineering*, Vol. 21, No. 12, 1995, pp. 929-944.

[OUWE06] Ouwerkerk, J., Abran, A., 'Evaluation of the Design of Use Case Points (UCP)', MENSURA2006, Nov. 4-5, Shaker Verlag, 2006, Cadiz (Spain).

[SELA03] Sellami, A. and Abran, A., "The contribution of metrology concepts to understanding and clarifying a proposed framework for software measurement validation", 13th International Workshop on Software Measurement (IWSM), Montreal, Canada, Shaker Verlag, 2003, pp. 18-40.

[SELA05] Sellami, A., Abran, A., « Analyse d'une mesure de la taille fonctionnelle du logiciel selon le point de vue de la métrologie et application à la méthode COSMIC-FFP », Revue Génie Logiciel, Nanterre (France), No. 74, Sept. 2005, pp. 2-12.

[ZELK98] ZELKOWITZ M.V. & WALLACE D.R., *Experimental Models for validating Technology*, IEEE Computer, May 1998, pp. 23-31, URL: http://www2.umassd.edu/SWPI/science/r5023.pdf

[ZUSE97] Zuse H., *A Framework for Software Measurement*, Walter de Gruyter, Germany,Berlin, 1997, pp. 755