Software Measurement Standard Etalons: A Design Process

Adel Khelifi and Alain Abran

Abstract— Material measurement standard etalons are widely recognized as critical for accurate measurements in sciences and engineering. However, there is no measurement standard etalons yet in software engineering. The absence of such concept in software measurement can have a negative impact on software engineers and managers when they using measurement results in decision-making. Software measurement standards etalons would help verify measurement results and they should be included within the design of every software measure proposed. Since the process for establishing standard etalons for software measures has not yet been investigated, this paper tackles this issue and proposes a seven steps design process. A case study of this design process of a software measurement standard etalon is presented using ISO 19761: COSMIC-FFP.

Keywords— Software Engineering, Metrics, Software Measurement, ISO 19761, COSMIC-FFP, Standard etalon.

I. Introduction

MEASUREMENT in general is a mature science with a long tradition. In disciplines such as physics, chemistry and biology, it is a basic part of daily activities.

Measurement standards are designed to make life easier: for example, a liter is a well-known quantity around the world, and has exactly the same value in all countries. Similarly, from east to west and from north to south, the meter is the standard for length measurement: it is applied similarly everywhere and also has a single value.

According to the International Vocabulary of Basic and General Terms in Metrology [1] a standard etalon is "a material measure, measuring instrument, reference material or measuring system intended to define, realize, conserve or reproduce a unit or one or more values of a quantity to serve as a reference."

Using a standard etalon can improve competitiveness by reducing the cost of both manufacturing and market transactions: a producer does not need to reinvent the specifications or performance criteria incorporated in the standard, and can therefore concentrate resources elsewhere. Furthermore, a standard etalon can contribute to the propagation of innovations, and consequently enhance the economic benefit to be derived from them.

Consequently, it becomes relevant to develop, for both measurers and users of software measurement results, a system of references made up of software measurement standards. Measurement standards are essential elements for an adequate metrological structure, in that they provide software engineers with a common reference and give them greater confidence in the measurement process. Indeed, standards facilitate the realization of measurement results on common bases.

While it is difficult to determine the effect of measurements on software quality, it is clear that using standards of measurement would provide software measurers, developers and managers with much better indicators of that quality, as well as more time to react, and could reduce the number and seriousness of software failures. In the information technology domain, and more specifically in software engineering, concepts of units and etalons have seldom been used, and this is a symptom of the immaturity of the software measures themselves. Consequently, the field of software measurement is not yet mature enough to be recognized as having value in the daily practice of software development, nor for the purchase or sale of software products and packages.

It is difficult to develop measurement standard etalons. They are created through an iterative process in which each iteration represents an improvement over the previous ones, in terms of both accuracy and stability. Moreover, each iteration may span years, if not decades.

Up to now, some characteristics of software have made it challenging to measure (see Figure 1):

1. Software is an intangible product, and some doubt that metrology concepts are applicable.
2. Software is an atypical product when compared to other industrial products, in that it varies greatly in terms of size, complexity, design techniques, test methods, applicability, etc.
3. There is little consensus on specific measures of software attributes, as illustrated by the scarcity of international standard measures for software attributes, such as software complexity and quality.

Manuscript received December 28, 2007.

Adel Khelifi is with ALHOSN University, Abu Dhabi, P.O BOX 38772 UAE (phone: 00971-02-407-0723; fax: 00971-02-407-0799; e-mail: a.khelifi@alhosnu.ae).

Alain Abran is with the Universidad del Pais Vasco, San Sebastian - Donostia, Spain, on leave from the Department of Software Engineering and Information Technologies, École de technologie supérieure - Université du Québec, Montréal, Canada H3W 1T8 (e-mail: alain.abran@etsmtl.ca).

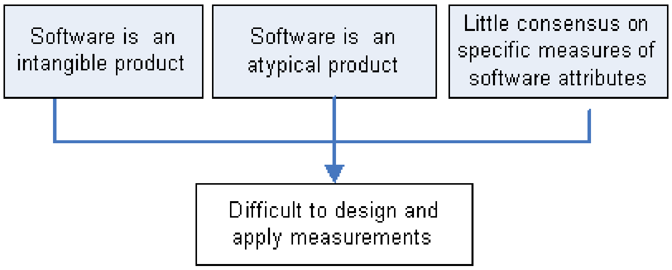

Fig. 1. Challenges in the design of software measures

Because of these challenges, some have claimed that software "metrics" are somewhat unique, and, as such, cannot be constrained to meet all the metrological properties as defined in the ISO document on metrology [1]. Currently, there is no standard etalon for software, but this fact does not imply that ones cannot be created. Indeed, there is a lack of documented attempts to do so, and the lack of a methodology for doing so for software. In this paper, we postulate that it is feasible to create a standard etalon for software and that a methodology for doing so could be designed.

If measurement reference material in the form of standard etalons were to be available to software practitioners, it could:

- be used as a common baseline for measurement;
- offer a point of reference for software measurers to verify their measurement results and their ability to measure the same reference material;
- allow measurers to use the related reference concept, and thus to speak at the same level.

The focus of this paper is the proposal of a design process for developing a standard etalon, and initially for a single type of software measures, that is for a software Functional Size Measurement (FSM) method.

The motivation for proposing an initial software measurement standard etalon for functional size is the need in this measurement community for a traceable and widely recognizable standard etalon which could be used as reference material in contractual agreements on one hand; on the other hand, it could be used in verification of software tools which being developed by both researchers and vendors attempting to automate this type of software measures.

The structure of the paper is as follows: Section 2 presents related work in the design of measurement standards in general, and in FSM in particular. Section 3 presents a proposal for a design methodology for a software measurement standard etalon. Section 4 presents its application on ISO 19761 – COSMIC-FFP. Section 5 presents a discussion and identifies further research issues.

II. Related work in the design of a measurement standard-etalon

A. Primary reference material, calibration and testing

A measure is first defined in terms of its objectives, a meta-model of the entity to be measured and the characteristics of the attribute to be measured. This definition is then realized by means of a measurement unit, a corresponding scale and the assignment of numerical rules [2, 3].

Next, to ensure that measurements are performed in a consistent manner, a base line is established as a primary reference (i.e. a standard-etalon).

Any measure can be compared with the standard-etalon by means of calibration and testing [4]. Calibration determines the performance characteristics of an instrument or the reference material used in a particular measurement with respect to the standard-etalon. There are three main reasons for calibrating an instrument:

1. To ensure that the instrument readings are consistent with other measurements.
2. To determine the accuracy of the instrument readings.
3. To establish the reliability of the instrument, i.e. that it can be trusted.

Reference procedures can be defined as measurement or analysis procedures which are thoroughly characterized and proven to be under control, and intended for the quality assessment of other measurement procedures for comparable tasks.

The uncertainty of the results of a reference procedure must be adequately estimated and appropriate for the intended use.

Reference procedures can be used, for instance, to validate other measurement or test procedures used for a similar task and to determine the level of uncertainty associated with them.

Uncertainty is a quantitative measure of the quality of a measurement result enabling the measurement results to be compared with other results, references, specifications or standards.

B. Design issues for the measurement of the software concept-entity

In the software engineering literature, measurement concepts are often defined in vague ways. For example, the term "metric" has several definitions [2, 3, 5] and the designers of software metrics have not yet embedded in their design the full set of measurement concepts that is embedded, and widely accepted, in the traditional field of metrology used extensively in the engineering disciplines. It has also been recognized by authors who have discussed frameworks for "metrics" validation that such frameworks are still incomplete [6, 7, 8, 9], with little theoretical basis and a lack of reference to metrology concepts and criteria. For instance, it has been observed that, in software engineering, most measurement proposals do not refer to any references (primary references or others), do not suggest any measuring instrument and do not design or adopt any measurement standard [10].

C. Software Functional Size Measures (FSM)

For illustrative purposes, a single type of software measures has been selected, that is, functional size. The key reason for this selection is that, of the numerous types of measures proposed for software, functional size measures (FSM) are currently the only ones to have developed a broad enough consensus to gain widespread recognition as international software measurement method standards.

FSM is "the approach to quantifying software in terms of the functionality it delivers to its users independently of the technical and quality aspects of its delivery. It provides a method of normalizing measures of productivity, speed of delivery, quality, etc. by providing a common measure of what is delivered which can be used to calculate unit values" [11].

The reader is reminded that Functional User Requirements are defined as "a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfill the user's needs. They exclude Quality Requirements and any Technical Requirements" [22].

ISO has developed a set of meta-standards with respect to FSM, that is, its ISO 14143 series, parts 1 to 6:

Part 1: Definition of Concepts

Part 2: Conformity Evaluation of Software Size Measurement Methods

Part 3: Verification of Functional Size Measurement Methods

Part 4: Reference Model

Part 5: Determination of Functional Domains for use with Functional Size Measurement

Part 6: Guide for use of ISO 14143 series and related International Standards.

In the specific domain of software FSM, four methods have been recognized as ISO international standards:

-ISO 19761: COSMIC-FFP [12].

-ISO 20926: Function Point Analysis (e.g. IFPUG 4.1, unadjusted function points only) [13];

-ISO 20968: Mk II [14]

-ISO 24570: NESMA [15]

In practice, the application of software functional measures requires knowledge in the specific software measurement method being used and sufficient experience in the interpretation of software artifacts. For instance, in the measurement process with the COSMIC-FFP method, the measurer must determine the following, from the available artifacts: software layers to be measured, software boundary, users, triggering events, functional processes, data groups and data movements. Should the documentation be complete and accurate, these measurement steps are easy. Unfortunately, in practice, the documentation is often incomplete, and, to measure software, the measurer has to supplement the information provided on some requirements which is either incomplete or ambiguous.

None of the four ISO-recognized FSM methods explicitly addresses the concept of a standard etalon while only COSMIC-FFP specifically specifies and documents the concept of a size unit.

The availability of a standard etalon for FSM would help improve the quality of FSM results on a practical level. Using a standard etalon can, therefore, help reduce the time spent in addressing inconsistency issues in measurement results and facilitate the verification and calibration of tools built to automate this measurement method in specific environments.

D. Related work in FSM

Use of case studies as training material

Up to now, the respective measurement communities for each of the four ISO-recognized FSM have mostly developed case studies as reference material for training purposes, and these are very specific in terms of teaching some peculiarities of each FSM method; however, they are not yet generic enough to be used as reference material for calibration and testing purposes.

These case studies suffer from a number of limitations:

- there is no normalized input to their design process;
- they have been drafted based on the judgments of experts within their own communities;
- they are limited in scope;
- they most often address only a limited number of measurement rules, sometimes in peculiar contexts.
- they cannot be used as generic reference material.

ISO work in ISO 14143

The ISO has indirectly recognized the need for reference material through its provision of reference input material for measurement: indeed, ISO technical report 14143-4 provides a set of Reference User Requirements (RURs) which were put together to provide FSM communities with material that could be used for convertibility studies across specific measurement methods. Such reference material could also be used to test some of the metrological properties of a specific measurement method, such as the accuracy, repeatability and reproducibility criteria quoted in ISO TR 14143-3.

However, ISO TR 14143-4 suffers from a number of important limitations. In its current state, ISO 14143-4 cannot be used to assess an FSM method against some standard reference points to determine whether or not it yields expected results in a given situation: in this standard, all the sets of RURs are described in a non-standardized textual format. There is, therefore, a great variation in the description of these RURs within a given set, and, of course, across sets.

In functional size measurement, the measurement process relies, generally, on its functional documentation [16]. It has been illustrated in Nagano et al. [17] that the quality of the documentation has an impact on both the quality of the measurement results and on the effort required to carry out the measurements. Several researchers [18] have noted that the software documentation is often either incomplete or obsolete, and even sometimes erroneous.

For instance, it was observed that distinct measurers produce different measurement results when they need to make assumptions (which will vary often from one person to another based, in particular, on their work experience) in the absence of complete or unambiguous requirements (of course, distinct developers implementing such incomplete and ambiguous requirements would produce distinct software designs and related software implementations).

None of the sets of 'reference user requirements' in ISO TR

14143-4 has been reviewed for quality control: trial uses both by experts and by beginners have highlighted a number of ambiguities and a lack of completeness, leading to different interpretations of these ambiguous functional requirements, and, of course, to various measurement results.

Related work on COSMIC-FFP

The topic of a standard etalon for ISO 19761 -COSMIC-FFP was initially discussed in [19] and initial drafts were documented in [20] where the main objective was the construction of a set of references for software measurements. It includes eight sets of functional user requirements covering three types of software; business applications, real-time system and hybrid system. Five of them come from the ISO 14143-4 technical report; they are Automatic Line Switching System, Hotel System Reservation, L-Euchre Application, SAGA System and Valve System Control. A FUR belongs to the Rice Cooker application. The last two sets of FUR belong to the training documents of the IBM-Rational company, and used with permission: they are C-Registration System and Collegiate Sports Paging System.

A limitation of this pioneering work is that it is an individual effort and does not benefit from international recognition or worldwide diffusion. Official international recognition of a standard etalon for software measurement would be of practical interest to both industry and researchers.

The work reported next builds on that in [20] and extends it to any FSM, and, by extension, potentially to any software size measure.

III. A DESIGN METHODOLOGY FOR AN FSM MEASUREMENT STANDARD ETALON

The challenge is how to design a standard etalon for software which is not a material product. The generic process described below is based on the lessons learned from the preparation of case studies for training purposes and from work done to explore the design of an initial draft version of etalons for the COSMIC-FFP method, as well from the work reported in [20].

This section presents a design process for developing a software measurement standard, including the following seven steps – see Figure 2.

1. Analysis and selection of candidate textual description of Functional User Requirements (FUR); the input is the literature survey of previous work on the design of a specific measurement method and available descriptions of FUR. However, these sets of FUR are most often available only in non-standardized textual format.

2. Identification and selection of quality criteria for the input to the measurement process. For FSM, the inputs are usually expressed in the form of textual descriptions of requirements, and related quality criteria are defined, for instance, in the IEEE standards on Specifications Requirements – IEEE 830. These quality criteria then become inputs to step 3.

3. Quality improvement of the set of FUR: this step consists in the transformation of the selected set of textual FURs into the selected specification language, and, in parallel, analysis of the quality of the requirements and correction of requirements defects (for instance, to remove ambiguities and inconsistencies in the requirements). An experienced measurer is required for this step and he must use the specification notation language selected by the organization he is working for.

If such a selection has not been made, a step must be added for such a selection of a specifications notation language.

The output of this step is then the set of FURs described in the selected notation specification language and which meet the specified quality criteria. This set of FUR is then the input for the development of the standard etalon – said differently, the output of this step becomes the input that is the measurand which will be measured and which will form the basis of the standard etalon.

4. Design of an etalon template for presenting the measurement process and measurement results. If such a template already exists for this type of measures, then this step can be skipped.

5. Initial measurement process: this step consists in the initial measurement of the requirements documented in the adopted specification notation by an experienced measurer to produce an initial draft of measurement results using the adopted output format for the standard etalon. The output of this step is the initial measurement results documented using the template for a standard etalon.

6. Selection of experts for verification: this step consists in the selection of a group of experts to review the initial measurement results. Ideally, these measurement experts should be internationally recognized by industry for their specific FSM expertise; of course, it would add credibility if these experts were also active participants in an ISO standardization program on software measurement.

7. Verification cycle: this step consists in the revision by expert measurers of the initial measurement results and correction of either the inputs (the requirements themselves if they were incomplete or ambiguous) or of the outputs (the measurement results).

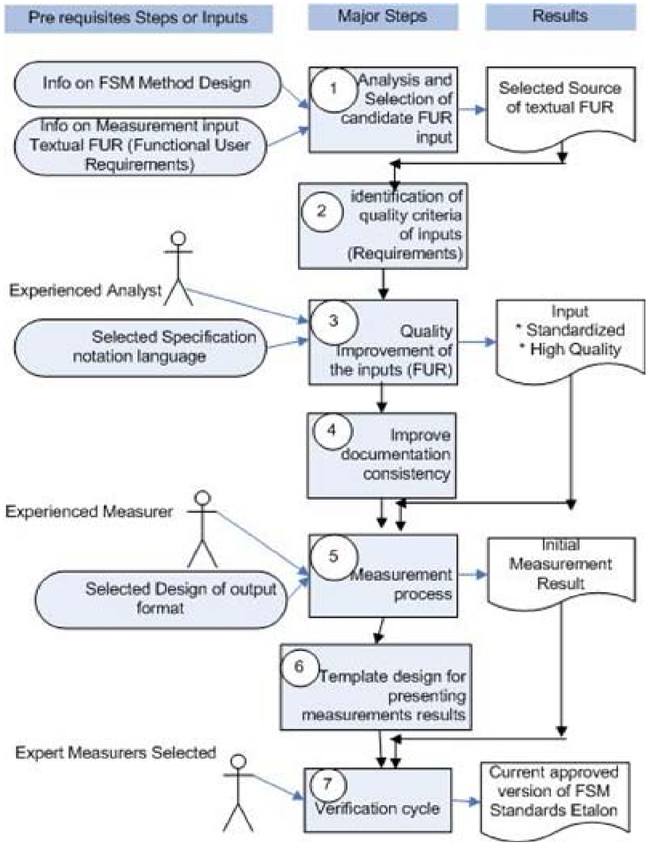

Fig. 2. A methodology to develop a software measurement standard

IV. The Methodology for a COSMIC-FFP Measurement Standard Etalon

This methodology for developing an FSM standard etalon is a generalization of the steps carried out in [20]. Of course, the modeling of these steps has been further refined. Its specific instantiation for COSMIC-FFP is documented next.

1. Analysis and selection of candidate FUR as input. This step includes the prerequisites to beginning the process of designing a standard etalon for COSMIC-FFP. In this specific instance, it consists of the output of the literature survey of previous work on the design lessons learned from COSMIC-FFP case studies, as well as on the identification of a set of candidate inputs for measurements. In this specific instance, the ISO work on FSM was selected (that is, ISO TR 14143-4 2000 − Reference User Requirements (RURs) [21]), since it contains an inventory of textual descriptions of requirements collected for measurement purposes.

Since the input to this step contains multiple sets of requirements, one specific set was selected as the basis for the work reported here, which was RUR B9 − Valve Control System (from ISO 14143-4).

2. Identification of quality criteria of the inputs (i.e. or the requirements).

The quality criteria selected as prerequisites were selected from the IEEE standard on software requirements, that is, IEEE 830.

3. Quality improvements of the inputs.

In ISO TR 14143-4, all the sets of RURs are described in a non-standardized textual format. There is, therefore, great variation in the description of these RURs within this specific B9 set. This is typical of most inputs for the measurement of the functional size of software, in particular when the measurements are taken early in the software life cycle. As a result, it is necessary to verify the quality and completeness of these requirements. The RURs are therefore analyzed, verified and improved using the quality criteria identified in the previous steps, that is the quality criteria from IEEE 830.

In this step, a specification language is selected as an input, and the selected set of textual FURs is transformed into a specification language. To improve the consistency of the documentation to be used as input to the FSM, the decision was made to adopt the UML notation for this research, such as use cases and sequence diagrams for the software to be measured. The UML Use Case diagram is a tool for representing the entire functionality of a system; a sequence diagram is a structured representation of software behavior as a series of sequential steps over time. Developing such diagrams can improve the comprehension of software functions and provide the measurer with more consistent and precise documentation as input to his measurement process.

This allows the measurer to have his measurement inputs documented in a consistent manner, which in turn allows him greater transparency in the intermediate steps of the measuring process and more repeatable results. For illustrative purposes, Figure 3 presents the sequence diagram for one of the case studies measured for the design of an initial version of a standard etalon.

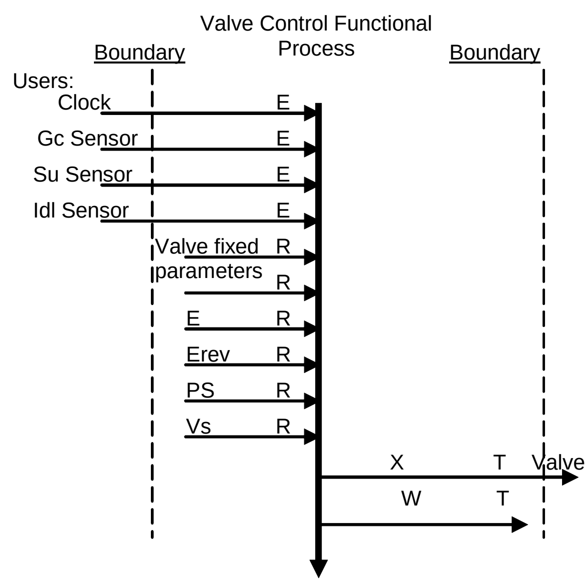

Fig. 3. Valve Control Application − Sequence Diagram

An analyst with expertise in UML notation carried out this step, which consisted of analyzing the textual description of the requirements and their transformation into UML notation, and, within this process, the correction of defects (for instance, to remove ambiguities and inconsistencies in the requirements). The outcome then is the verified set of Functional User Requirements to be measured, that is the measurand.

4. Design template for presenting the measurement results

The next prerequisite for a major step is the selection or design of a template for presenting the measurement process and measurement results: since there had already been documented case studies for COSMIC-FFP, these were reviewed and tailored for the purpose of documenting the intermediate steps of the measurement process, as well for the outcome in terms of measurement results. An example of a template for a COSMIC-FFP standard is presented in Box 1. This template is an evolution of the reports developed by the COSMIC Consortium and the GELOG [23] for documenting case studies.

1. Overview
   1.1 Introduction
   1.2 Measurement viewpoint, purpose and scope
2. Requirements as documented in ISO 14143-3-4: 2000
   2.1 Context
   2.2 Input
   2.3 Output
3. COSMIC-FFP measurement procedure
   3.1 Identification of layers
   3.2 Identification of users
   3.3 System boundary
   3.4 Identification of triggering events
   3.5 Identification of data groups
   3.6 Identification of functional processes
4. Identify data movements
   4.1 Message sequence diagram
   4.2 List of data movements
   4.3 Observations on the requirements' clarity
5. Analysis of measurement results
6. Summary, including observations
7. Questions & answers

Box 1: Template for a COSMIC-FFP standard etalon

5. Initial measurement

The initial measurement was performed, by an experienced measurer, of the requirements documented in the adopted specification notation to produce an initial draft of measurement results. The quantitative measurement results for this case study are summarized in a pie chart, with the percentage of COSMIC-FFP data movement types of the measurement result for the case study (Figure 4), while the detailed inputs and outputs are documented with the output format selected (that is, Box 1).

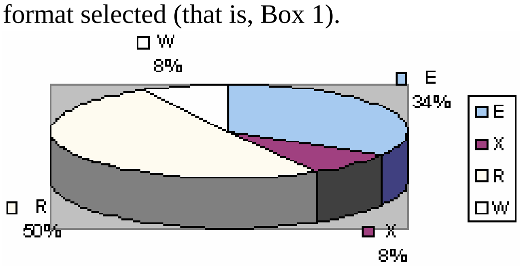

Fig. 4. Percentage of COSMIC-FFP data movement types

6. Selection of experts

In this step, a group of experts was selected to review the initial measurement results; ideally, these measurement experts should be internationally recognized by industry for their specific FSM expertise; of course, it would add credibility if these experts were also active participants in the ISO standardization program on FSM. Ideally, the design of standards is an activity which must be undertaken at the international level by groups of experts from several countries in order to obtain a broad consensus. The ISO organization represents the most adequate framework for this type of activity. The selection of experts for the draft COSMIC-FFP standard etalon was made through the Software Engineering Research Laboratory contacts. It included international experts in software measurement within the COSMIC group, a group of international volunteer experts in software measurement. Some of these experts were also members of WG12, an ISO working group specializing in software FSM. However, this work was not done in an official context, and the credibility of the measurement outcomes is derived from their individual expertise, and not from an official international process recognized by national institutions.

7. Verification cycle

This step constituted a review of the initial measurement results and correction, even of the requirements themselves if they were incomplete or ambiguous. The final output was then the currently approved version of a standard etalon for COSMIC-FFP. It is to be noted that, for traceability purposes, the output in software measurement must include both the inputs and the outputs of the measurement process for establishing the standard etalon.

In summary, the end-result of the design of a standard etalon for software FSM with the COSMIC-FFP method consists of a detailed report using a template documenting both the inputs and the outputs of the measurement process on a set of software FURs.

The verification process embedded within this design methodology is highlighted below –see Figure 5- and involves:

1. Individual verification;
2. FSM experts' verification process;
3. Systematic verification by the COSMIC measurement practice committee.

The iterative verification process is highly relevant at the international level; in practice, this verification process will go through an iterative cycle

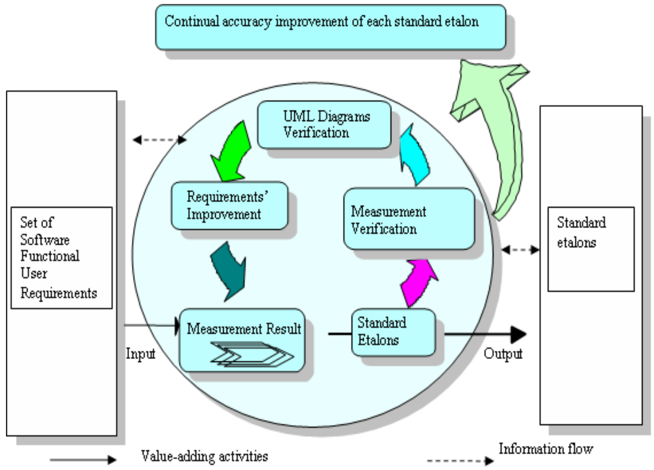

Fig. 5 Iterative verification process for a software standard etalon

V. DISCUSSION

It can be observed that the process presented in this paper for designing a software standard etalon did not produce per se a 'material' standard etalon, but rather, as mentioned as well in the International Vocabulary of Basic and General Terms in Metrology [1], "reference material or measuring system intended to define, realize, conserve or reproduce a unit or one or more values of a quantity to serve as a reference".

The development of a standard etalon for software measurement could have a far-reaching impact: for instance, many standard etalons, such as the meter standard or kilogram standard etalon, contribute in the management of many aspects of our daily life.

From our point of view, the use of software measures should be integrated in a complete process of verification, where measurements assess attributes which are related to the main purpose of the software and enable us to check the credibility of the results. In order to optimize a software measurement application, measurers have to know the 'why' and the 'what' of the measurement itself. The use of models in software measurement is a predetermining factor of measurement consistency.

In this paper, we presented a process for developing a standard etalon for software measurement and illustrated it using ISO 19761 – COSMIC-FFP. The application of the COSMIC-FFP measurement method by experts in software FURs generates the measurement results. It is the consensus among measurement result experts that defines the quality of a standard etalon for the result. The verification of every part of the standard etalons by recognized experts and COSMIC members provides the standard etalon with greater accuracy. The addition by measurers or software engineers of UML diagrams, use cases and sequence diagrams, and their verification by UML developers, further enhances the software functionalities by providing greater understandability, accuracy and completeness. This allows the measurers to re-analyze the measurement results and make other improvements if necessary.

Meanwhile, it is important that the software measurement community comes to appreciate that the development of a standard for the measurement of software may take many decades. For instance, it took two centuries for the definition of the meter to become established.

In conclusion, we, as designers of software measures, must learn how to build standards for software and accept that, as for any other standard etalons in physical sciences, initial software standard etalons will require improvements over time to provide the software engineering community with progressively more accurate standard etalons.

REFERENCES

[1] International Vocabulary of Basic and General Terms in Metrology, BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML, International Organization for Standardization, Geneva, 1993.
[2] J.-P. Jacquet; A. Abran,. "From Software Metrics to Software Measurement Methods: A Process Model," Third International Symposium on Software Engineering Standards, ISESS'97, Walnut Creek, CA. June 2-6, 1997.
[3] A. Abran, 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.
[4] P. Howarth, "Metrology – in Short", 2nd ed. Vol. 1. Lyngby: EUROMET, 2004. EUROMET project no. 595. 18 Sept. 2004, www.euromet.org/docs/pubs/docs/metrology_in_short_2nd_edition_may_2004.pdf?.
[5] J. Bieman, N. Fenton, D. Gustafson, A. Melton, L. Ott., "Fundamental issues in software measurement", in Melton, A. ed. Software Measurement: Understanding Software Engineering, International Thomson Publishing (ITP), 1996, 39-74.
[6] C.F. Kemerer, "An Empirical Validation of Software Cost Estimation Model", Communications of the ACM, 30 (5). 416-429.
[7] D. Kafura, J Canning, "A Validation of Software Metrics Using Many Metrics and Two Resources", 8th Intl. Conf. on Engineering, (Washington, D. C, 1985), 378-385.
[8] B. Kitchenham, S.L. Pfleeger, N. Fenton, "Towards a Framework for Software Measurement Validation", IEEE Transactions on Software Engineering, vol. 21, no. 12, 1995, pp. 929 -944.
[9] A. Sellami, A. Abran, "The contribution of metrology concepts to understanding and clarifying a proposed framework for software measurement validation", 13th International Workshop on Software Measurement – IWSM 2003, Montréal (Canada), Shaker Verlag, Sept. 23-25, 2003, pp. 18-40.
[10] A. Abran, A. Sellami, W. Suryn, "Metrology, Measurement and Metrics in Software Engineering", METRICS 2003, Sydney, Australia, September 2003, IEEE Computer Society, Los Alamitos, 2003.
[11] UKSMA, "Functional Size Measurement Methods," 2005, http://www.uksma.co.uk/?action=35&what=35&type=0.
[12] ISO/IEC 19761:2003, Software Engineering – COSMIC-FFP – A Functional Size Measurement Method, International Organization for Standardization – ISO, Geneva, 2003.
[13] ISO/IEC 20926: 2003, Software Engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual International Organization for Standardization – ISO, Geneva, 2003.
[14] ISO/IEC 20968: 2002, Software Engineering – Mk II Function Point Analysis – Counting Practices Manual, International Organization for Standardization – ISO, Geneva, 2002.
[15] ISO/IEC 24750: 2005, Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization – ISO, Geneva, 2005.
[16] A. Abran, "Analyse du processus de mesure des points de fonction", département du génie électrique et de génie informatique, PhD thesis, Ecole Polytechnique de Montréal, Montréal, 1994, p. 405.

[17] S.-I. Nagano, K.-I. Mase, Y. Watanabe, T. Watahiki, S. Nishiyama, "Validation of Application Results of COSMIC-FFP to Switching Systems," ACOSM2001, 17 Nov., 2001, Australian Government – Department of Employment and Workplace Relations. 6 Jan., 2004, www.gelog.etsmtl.ca/publications/pdf/343.pdf.

[18] J.-M. Desharnais, A. Abran, "Applying a Functional Measurement Method: Cognitive Issues," 11th International Workshop on Software Measurement (Montréal, Québec, 2001), Shaker Verlag, 2001, pp. 26-50.

[19] A. Khelifi, A. Abran, L. Buglione: "A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)," 14th International Workshop, IWSM 2004, Berlin, Germany, November 3-5, 2004

[20] A. Khelifi, "A Set of References for Software Measurement with ISO 19761 (COSMIC-FFP): an Exploratory Study", Département du génie logiciel, PhD thesis, École de Technologie Supérieure, Montréal, 2005, p. 495.

[21] ISO/IEC TR 14143-4: 2000 Information Technology – Software Measurement – Functional Size Measurement – Part 4: Reference Model, International Organization for Standardization – ISO, Geneva, 2000.

[22] ISO/IEC 14143-1: 1998. Software Engineering – Functional size measurement – Part 1: Definitions of Concepts, International Organization for Standardization – ISO, Geneva, 1998.

[23] GELOG. (2004), "Valve Control System Case Study", Ecole de Technologie Superieure. Montréal, Canada:http://www.gelog.etsmtl.ca/cosmic-ffp/casestudies/valvecontrol_release_january_2006.pdf

Doctor Adel Khelifi was born in Tunisia in 1967.
Dr. Khelifi holds a PhD in software engineering from the École de Technologie Supérieure (ETS) – Université du Québec (Montréal, Canada (2005), a Master in Management Information Systems from the University of Quebec, Canada (2001), Systems Analyst Diploma from the National Centre of Computer Science, Tunisia (1995) and a Bachelor degree in Industrial Engineering from Annaba University, Algeria (1991).

He is Assistant Professor in the Software Engineering department and IT Director at ALHOSN University, UAE. He has held impressive past careers, most recently working as a lecturer for the Engineering School of Technology in Canada and previously working for United Nations MSF in Canada, Ministry of Relations with Citizen and Immigration in Canada and Ministry of Finances in Tunisia. Currently, he is involved in developing the Software Engineering courses' content including software quality, software testing and software maintenance for the students.

As a Canadian ISO member in Software Engineering, Dr. Khelifi is contributing in developing software measurement standards.

Doctor Alain Abran was born in Canada in 1949.
Dr. Abran holds a Ph.D. in Electrical and Computer Engineering (1994) from École Polytechnique de Montréal (Canada) and master degrees in Management Sciences (1974) and Electrical Engineering (1975) from University of Ottawa.

He is a Professor and the Director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ETS) – Université du Québec (Montréal, Canada). He has 15 years of experience in teaching in a university environment as well as more than 20 years of industry experience in information systems development and software engineering. His research interests include software productivity and estimation models, software engineering foundations, software quality, software functional size measurement, software risk management and software maintenance management.

Dr. Abran is currently a member of the IEEE, a Marie-Curie International Fellow, a co-editor of the Guide to the Software Engineering Body of Knowledge project and a Co-chair of the Common Software Measurement International Consortium (COSMIC).