A System of References for Software Measurements with ISO 19761 (COSMIC-FFP)

Adel Khelifi, Alain Abran, Luigi Buglione

Ecole de Technologie Supérieure - ETS

1100 Notre-Dame Ouest,

Montréal, Canada H3C 1K3

adel.khelifi.1@ens.etsmtl.ca, aabran@ele.etsmtl.ca, luigi.buglione@computer.org

Abstract:

Software measurement is still emerging as a field of knowledge, and, most often, traditional quality criteria of measurement methods such as repeatability, reproducibility, accuracy and convertibility are not even investigated by software measurement method designers. In Software Engineering, the Functional Size Measurement (FSM) community has been the first to recognize the importance of such quality criteria for measurement, as illustrated in the recently adopted ISO document 14143-3; these criteria represent, however, only a subset of the metrology criteria which includes, for instance, measurement units and internationally recognized measurement references (e.g. 'etalons'). In this paper, a design for building a set of normalized baseline measurement references for COSMIC-FFP (ISO 19761), the 2nd generation of FSM methods, is proposed. The goal is to design, for the first time in Software Engineering, a system of references for software FSM methods.

Keywords

System of References, Software Engineering Measurements, Functional Size Measurement, Cosmic-FFP, ISO/IEC 14143, ISO/IEC 19761

1 Introduction

1.1 Measurement concepts and relevance

"*Man is the measure of all things"* (Protagoras 485 BC). Measurement is an integral part of any human activity: social, economic, industrial, academic, environmental, medical, etc. Ubiquitous in our daily activities to provide an objective vision on quality, measurement has become a foundation for industrial, scientific and social development.

Measurement plays an important part in science and in the engineering disciplines, as well as in our daily lives.

Can we imagine civil engineers constructing buildings and bridges without measures being taken before and after the construction phase, people buying clothes without measuring their size, spectators attending events without knowing their duration, or, more dangerously, pharmacists filling drug prescriptions without measurements? By contrast, ‘no measurement used’ represents the state of practice in software development, even though there exists a large body of knowledge about software measurement [2] [9] notwithstanding the fact that the domain is relatively new and not yet mature.

Nowadays, organizations are in a situation where they must develop or renovate their software. Measurement can be a major analytical tool for better understanding and controlling the development and maintenance of software costs. For implementing changes prudently, measurement is of considerable importance, as it is for controlling expenses, deadlines and performances. Thirty years ago, software measurement was an area for creative thinking confined to university researchers and to Industrial Engineering, one of the first papers on the subject probably being Rubey et al. [29]. However, since the end of the ’70s, software measurement has been widely recommended. According to Zuse [31], the concept of measurement is discussed or referred to, in one way or another, at a majority of Software Engineering conferences.

The need for measurement has been also explicitly recognized in the IEEE’s own definition of Software Engineering [13]: "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software." The term ‘quantifiable’ positions measurement as an integral part of Software Engineering and not simply an add-on.

According to Fenton and Pfleeger [11], software is a physical entity which can be measured by its size, since physical objects are easily measurable. Also according to these authors, the software size measurement operation should be easy. However, software size measurement presents many difficulties, and this is because the concepts of effort, functionality and complexity related to software measurements have neither agreed upon boundaries nor precise definitions. Fenton & Pfleeger suggest that software size can be described by three attributes: length, complexity and functionality. In this paper, we focus on the measurement of the functionality attribute of software size, known as software Functional Size Measurement (FSM).

Even though over the past thirty years there has been considerable progress in Software Engineering, including a large number of proposals for measures and metrics, measurement is still not widely used. Software measurement is still emerging as a field of knowledge and, most often, traditional quality criteria of measurement methods, such as repeatability, reproducibility, accuracy and convertibility, are not even investigated by software measurement designers.

In Software Engineering, the FSM community has been the first to recognize the importance of such measurement criteria, as illustrated in the recently adopted ISO/IEC TR 14143-3 [17]. These criteria, however, represent only a subset of the metrology criteria which include, for instance, measurement units and *'etalons'* (an international material standard used for traceability to internationally recognized measurement references).

There is not yet a system of references for software measurement, and it is not even discussed in the Software Engineering literature. Even the FSM methods recognized by the ISO do not yet have such a system, with the exception of a few illustrative case studies.

A system of references for software FSM would provide a professional framework for software measurers and contribute to the evolution of Software Engineering measurement. International official recognition of a system of references for Software Engineering measurement is of particular interest to both industry and researchers. The goal of this research project is to design, for the first time in Software Engineering, a system of references for software FSM results.

In this paper, a design is proposed for building a set of normalized baseline measurement references for one specific FSM method, that is, COSMIC-FFP (ISO/IEC 19761:2003 [22]), the 2^{nd} generation of FSM methods. Such a design could contribute to providing measurement results, the accuracy of which would be directly traceable to it.

1.2 Research motivation and research issue

The motivation for this research is the need for traceable and widely recognized measurement references in software measurement, as in any other human endeavour with respect to measurement. The following set of simple questions highlights some of the major general measurement-related issues:

- How can you be sure the air temperature is 70°F, by watching the weather bulletin on TV?
- Is this the exact temperature? What temperature is being presented?
- Does it include the wind factor? Was it measured near the coast, or at high attitude?

In mature areas of measurement, the answers to such questions are derived by traceability to widely recognized measurement references with well-documented measurement properties. In mature disciplines of engineering, a result of measurement is accepted if it is traceable to one or more references. We always wonder who is providing this result and to what it is referenced. A measurement result always relates to a system of references.

A number without a system of references is not a measurement result, but simply an assertion. Similarly, an evaluation of cost only applies to that which expresses it, at the time when it expresses it and according to its system of reference. Just look at the endless discussions and debates that cannot come to a mutually agreed upon conclusion if the speakers' systems of reference are different.

That is why, in any discussion, it is necessary to fix the references used, otherwise it becomes difficult to reach conclusions acceptable to all participants. An example of such a system of measurement references is provided by the International System of Units. Such a measurement system provides a framework of international coherence and provides universal access to knowledge, good practices, feedback from experience and reference documents.

The development of an international software measurement system of references documented in a widely recognized system can have a far-reaching impact; for instance, many measurement reference systems within the regulatory and monetary systems contribute to managing some fundamental aspects of our daily lives.

Section 2 introduces related work about standardization in measurement, including the measurement of software functional size, as well as our research goals and objectives. Section 3 discusses related metrology concepts, the liaison method for recognizing references and the design of the research methodology. Section 4 presents the expected research outcomes in terms of design and verification criteria. Finally, section 5 summarizes the industrial impact of such a reference system, as well as suggestions for further work.

2 Related Work

2.1 Background on measurement standards

Measurement is one of the key concepts contributing to the maturation of engineering disciplines. Measurement, an intellectual construction, is ubiquitous in most human beings' activities. For example, the earliest measurements of distance were derived from the lengths of body members. By 3000 years ago BC, there were well-documented standard measurements for the exchange of goods among cities and nations. While there has been a diversity of standards for measuring the concept of distance, it took over 4800 years following the height of (demise of?) classical Egyptian civilization for the design and acceptance of a standard measurement "*for all nations and all times*", as defined in the Metre Convention [8] and which we now know as the meter.

In the late 19th century, other international standards for measures were created:

- the kilogram as the unit of mass – equal to the mass of the international prototype of the kilogram;

- the second as the unit of time – currently equal to the duration of 9,192,631,770 periods of radiation corresponding to the transition between the two hyperfine levels of the ground state of the c? sium 133 atom;
- the Kelvin as the unit of thermodynamic temperature – currently equal to the fraction 1/273.16 of the thermodynamic temperature of the triple point of water [10].

2.2 Functional Size Measurement (FSM)

Allan Albrecht [5] from IBM identified in the '70s the need for a FSM method independent of the programming languages and techniques used to develop software. Therefore, he designed a method referred to as Function Point Analysis (FPA). Albrecht's design was based on the functionalities delivered to users. From his initial FPA method, subsequent improvements have been proposed over the years, including the one by the Common Software Measurement International Consortium (COSMIC[1]). Indeed, FSMs can be applied early in the life cycle, helping to build estimation models for calculating the effort required to develop the software; of course, they can also be applied at the end of the development phase, helping managers to build software productivity models, as well as to compare the productivity of two software projects using such measurement results.

The initial Albrecht FPA method led, however, to a number of distinct interpretations, as discussed in [1], [27] and [28]. In addition, researchers have documented a number of theoretical weaknesses [1], [26] and [27], as well as a lack of generalization across functional domains.

2.3 The ISO 14143 series of FSM standards

When FPA was proposed in the mid '90s as a candidate ISO standard, the ISO experts selected a more encompassing strategy to address some of the fundamental measurement requirements for the acceptance of FSM methods by the international standardization community. ISO 14143 was developed by ISO Working Group 12 (JTC1/SC7/WG12), and is now a six-part project providing an internationally accepted set of standards and technical reports describing the concepts of interest to designers and users of FSM methods:

---

[1] http://www.cosmicon.com

- ISO 14143-1 Part 1: Definition of Concepts [15] – This part of ISO/IEC 14143 defines the fundamental concepts of FSM, promoting consistent interpretation of FSM principles.
- ISO 14143-2 Part 2: Conformity Evaluation of Software Size Measurement Methods [16] – This part of ISO/IEC 14143 was developed to provide a process for checking whether or not a Candidate FSM method conforms to the provisions of ISO/IEC 14143-1:1998. The output from this process can assist prospective users of the Candidate FSM method in judging whether or not it is appropriate to their needs.
- ISO TR 14143-3 Part 3: Verification of FSM Methods [17] – This part verifies whether or not an FSM method meets the quality characteristics of a measurement method, which are repeatability and reproducibility, accuracy, convertibility, discrimination threshold and applicability to functional domains. It defines various methods by which the usefulness of a method can be determined.
- ISO TR 14143-4 Part 4: Reference Model [18] – This part provides standard Reference User Requirements (RURs). Its purpose is to assess an FSM method against some standard reference points to determine whether or not it yields expected results in a given situation.
- ISO 14143-5 Part 5: Determination of functional domains for use with FSM [19] – This Technical Report describes the properties and characteristics of functional domains, and the principle procedures by which characteristics of FURs can be used to determine functional domains
- ISO TR 14143-6 Part 6: The guide for the use of the ISO/IEC 14143 series and related international standards [20] is currently under development, at the WD stage.

It must be stressed that the ISO 14143 standard series does not define an FSM method, but presents the characteristics for a measurement method to be recognized as an ISO FSM method.

Once the generic ISO FSM standards had been adopted, four specific FSM methods were submitted to ISO, demonstrating their conformity to the mandatory features expressed in ISO 14143-1, and were recognized as international standards:

- ISO 19761: 2003 – COSMIC-FFP - An FSM method [22];
- ISO 20926: 2003 – Function Point Analysis [23] (Note: only the ‘unadjusted’ portion of FPA is recognized as conforming to ISO 14143-1);
- ISO 20968: 2002 – MKII Function Point Analysis [24], from UKSMA;

- ISO 24570: 2004 – NESMA FSM method version 2.1 [25] (to be published, current status DIS[2])

2.4 ISO 19761 standard (COSMIC-FFP)

In this paper, the COSMIC-FFP standard [22] has been selected for illustrative purposes; this FSM method is referred to as a 2nd generation FSM method and addresses some of the major weaknesses found in 1st generation FSM methods, such as:

- practical limitations such as weak relevance for many software types (e.g. real-time software and multi-layered software as in telecom applications and operating systems);
- theoretical weaknesses, including, for instance, mathematical operations related to the numerical scale types;

The COSMIC-FFP method is based on the analysis of software FURs, broken down into subprocesses and data movements. In the COSMIC-FFP method, software users include all types of users who exchange data with the software, including human beings, equipment and other software; 1st generation FSM methods, by comparison, which focus on human users, are often not even present in much embedded software which interacts only with engineering devices such as sensors and monitoring equipment. It should be noted that the definition of users in COSMIC conforms to ISO 14143-1.

The COSMIC-FFP measurement method involves applying a set of models, rules and procedures to a given piece of software as it is perceived from the perspective of its FURs. The result of the application of these models, rules and procedures is a numerical value representing the functional size of the software, as measured from its FURs. In COSMIC-FFP, the symbol **Cfsu** (**C**OSMIC **f**unctional **s**ize **u**nit) represents the quantitative value of the software functional size.

Last, but not least, the COSMIC-FFP method is based on solid theory and decades of international experience. "*It has been designed from the outset both to comply with the ISO standard for FSM (ISO 14143) and to be compatible with modern ways of specifying requirements (e.g. use cases and prototyping)*" [30]. This measurement method is applicable to many software types. It recognizes that modern software development uses components in various layers within software architecture, so it is possible to measure software layers that other methods cannot.

[2] Refer to http://www.iso.ch/iso/en/widepages/stagetable.html with the codes identifying the approval stage within ISO committees.

2.5 Research goal and objectives

A system of references is a major input in decision-making models: it is required to organize, choose, communicate and evaluate the necessary attributes of software FSMs. This research project is aimed at developing a 1st generation of a system of references for FSM methods, on the basis of the metrology body of knowledge.

Two specific objectives are related to this research issue and will be pursued here:

- How to design a system of references for software FSM;
- How to make this system of references available to the practitioner community to develop and implement professional measurement practices in Software Engineering.

3 Research methodology

As the design of a system of references in Software Engineering had not yet been tackled, a key intermediate deliverable of this research has been the identification, within the domain of metrology knowledge, of the key concepts and techniques required to do so.

For the design of a system of FSM references, the “liaison concept’ is the most relevant for ensuring the traceability of the initial unique universal measurement standards, and then the traceability of individual measurements to these unique standards. This section explains related metrology concepts and how to specify, interpret and link them, and how to add a measurement reference to the system.

3.1 Related metrology concepts

Metrology and its standards facilitate the exchange of goods, support production automation, increase product quality, increase consumer trust and improve living standards. The contributions of metrology have been significant to the development of international trade and the reduction of technical barriers to exchanging goods. Metrology is not only a particular discipline of the physical sciences, but it also forms the basis of many of our daily tasks [10].

The process model in Figure 1 illustrates that measurement results come from a measuring instrument, which is calibrated (or not) with quantities and units, uses a standard of reference and has some characteristics which limit its field of application.

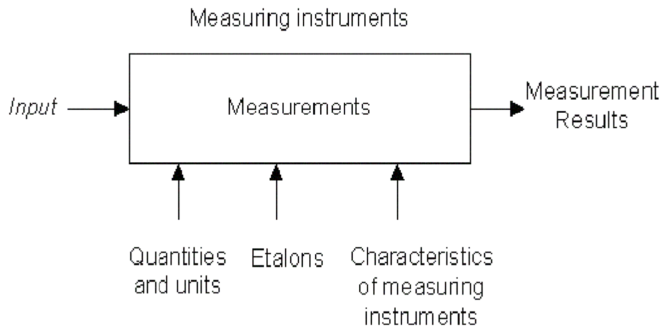

Figure 1: Process model of the categories of metrology terms [4]

While these concepts are well known in several fields, they are most often not discussed in Software Engineering: many of them are not embedded in the practices of software measurement designers, nor of Software Engineering measurers themselves. Metrology-related concepts and terminology have only been introduced recently into the ISO Software Engineering community; similarly, only a few researchers [3] have focused on metrology concepts as applied to Software Engineering standards, including specific methods of measurement. A subset of these metrology-related concepts was first accepted in ISO 14143-3, and then in the ISO 15939 standard for Software Process Measurement [21]. More recently, the metrology terminology has been introduced into the development of the next version of the ISO series for the measurement of the quality of software products (ISO series 25000 in JTC1/SC7/WG6).

We investigate next how the use of some of these metrology concepts in the design of Software Engineering measurements could contribute to developing new measurement-related knowledge for both researchers and practitioners. Particular emphasis is devoted to knowing how to develop *etalons* for software measures. Of course, software being a complex intellectual construct, it cannot be taken for granted that software-related *etalons* are simple. Quite the contrary, software as an intellectual product can be quite complex, even within a dimension such as size, and building corresponding measurement standards for such an apparently simple concept can be quite challenging; for instance, it would be expected that a single measurement standard could not yet be achieved, and that most probably it requires a complete system of references with examples of measurement results from software from many distinct functional domains.

3.2 Liaison method for recognized references

A key challenge for this project was to figure out how to build measurement standards (e.g. etalons).

The International Vocabulary of Metrology (VIM) [14] documents a number of different types of measurement standards, ranging from international standards to secondary standards to traveling standards – Table 1. The VIM also documents a number of characteristics for the conservation of a measurement standard: traceability, calibration, reference material and certified reference material. Each of these standards-related concepts can be considered as design requirements in the establishment of a system of measurement references.

(Measurement) Standard Etalon	**Conservation of a (Measurement) Standard**
International (Measurement) Standard	Traceability
National (Measurement) Standard	Calibration
Primary Standard	Reference Material (RM)
Secondary Standard	Certified Reference Material (CRM)
Reference Standard	
Working Standard	
Transfer Standard	
Travelling Standard	

Table 1: Detailed topology of Measurement Standards/Etalons

Once these requirements are identified, an approach must be found to tackle them. The Liaison Method as documented in [7] was selected as a relevant approach for the implementation of measurement standards (etalons). More specifically, the liaison of a measurement in a system of references must allow for the linking of any measurement result to one or more recognized measurement references. The method involves four steps, as presented in Table 2.

The four steps of the measurement Liaison Method, as adapted from [7], are:

1. Measurement specification. The optimal specification of a software measurement type requires:
    - descriptions and analyses of existing measurements for this type of software;
    - description of the transitions between measurements;
    - minimal description of the software environment;
    - follow-up allowing better definition of measurement changes.
2. Measurement interpretation compared to the system of reference. To interpret measurement results, the measurer draws from the body of knowledge of his professional environment and his personal experience. This interpretation depends on the system of references and many elements of the measurement process.

3. Liaison with the system. The liaison binds a measurement result to one or more references and then gives the measurement the corresponding name. This is done by technical reasoning, which is in the same order as measurement interpretation. The liaison is a flexible system that requires the study of the resemblance between a measurement and the references. To analyze this resemblance, we can use statistical concepts and methods such as multidimensional analysis. There exist both simple and multiple liaisons.
4. Addition of a new reference. If a measurement is far from all the references defined previously, we can add it to the system as a new reference. It is always possible to do this, as long as we do not affect system coherence. In the same way, it is possible to announce the existence of new types. However, in order to avoid confusion, any proposal for the creation of a new reference has to go through a preliminary detailed study and be argued in order to maintain general coherence.

3.3 Research steps

The following steps were identified as required to design an FSM System of References.

- **Step 1**: We use some of the key outputs derived from the ISO standardization work on FSM, that is, a specific measurement method adopted as ISO 19761(COSMIC-FFP).
- **Step 2**: We use some of the key outputs derived from the ISO standardization work on FSM; in particular, both the verification procedures from ISO TR 14143-3 and the RURs documented in ISO TR 14143-4, none of which, to our knowledge, have yet been used by industry or by researchers.
- **Step 3**: We integrate some of the concepts from three different tracks, namely Metrology concepts, International System of Units definitions and a methodology described in the field of pedology (the scientific study of the soil), for the definition of a '*Démarche de rattachement*', which is referred to here as a liaison method. This method will, of course, be transposed and adapted to the software measurement context.

Figure 2 presents these steps within our initial approach to the design of a System of References.

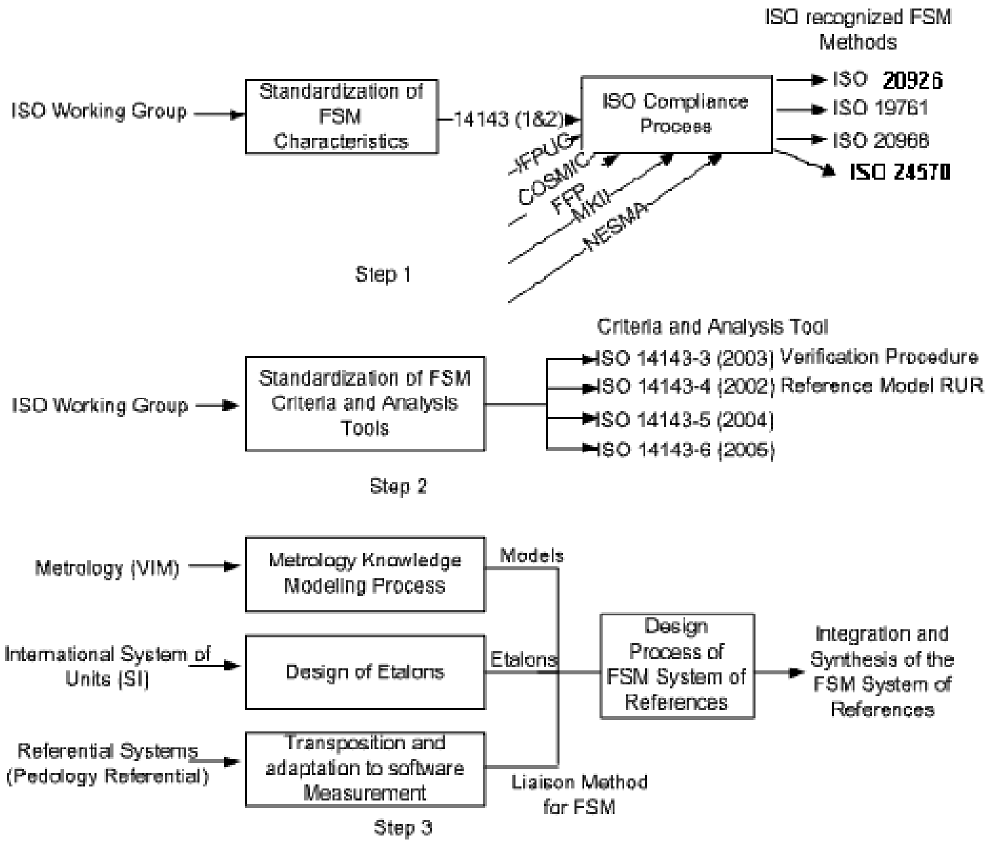

Figure 2: Initial approach to the design of an FSM System of References

These three sets of inputs are used next for the design of the System of References itself. The output of this design process will be the integration and synthesis of a system of references for FSM using COSMIC-FFP as a case study.

4 Design of the System of References

The design process for our FSM System of References includes both a verification process to ensure the quality of the measurement results, and a catalogue (i.e. repository) to keep track of measurement results and their traceability to the inputs for these measurement results.

4.1 Verification process

The application of an FSM method still currently depends on individual measurers' interpretations, either self-learned or derived from particular cases provided by a diversity of trainers.

Almost perfect measurement results would require a number of concurrent measurement conditions, such as:

- a perfect measurement method;
- a perfect measurement process;

- a perfect entry to the measuring process which is not affected by any 'noise'[3].

If such concurrent conditions existed, then the measurement results under perfect conditions would be the same. However, in Software Engineering practices, such ideal measurement conditions are seldom observed, the four main causes for differences in measurement results being:

- errors in the manual process of measuring;
- imperfections in the automatic measurement tool [12];
- defects in the measurement method itself;
- imperfect entries to the measurement process.

Of course, each of these main causes can be broken down into one or more subcauses, depending on the circumstances or on the context of measurement.

To build an optimal system of references would require that all perfect measurement conditions indicated above be present. The building of a system of references will necessarily be iterative for the progressive elimination of most of the sources of errors. Since this has not yet been done in Software Engineering measurement, it would be unreasonable to expect to produce perfect measurement results at the first attempt: several iterations will be required to eliminate the causes of errors.

To populate the catalog initially, some concurrent redundant measurement procedures will be carried out, such as:

- several people taking several measurements of the same RUR, in order to detect manual errors;
- several automated measurements being carried out with an experimental prototype of the automation of COSMIC-FFP in the Rational Rose environment [6].

In order to analyze and compare the measurement results of the same RUR, results will be verified using the procedure and criteria specified in ISO 14143-3: repeatability and reproducibility, accuracy, convertibility, discrimination threshold and applicability to functional domains. Figure 4 illustrates this verification process.

---

[3] In a statistical meaning.

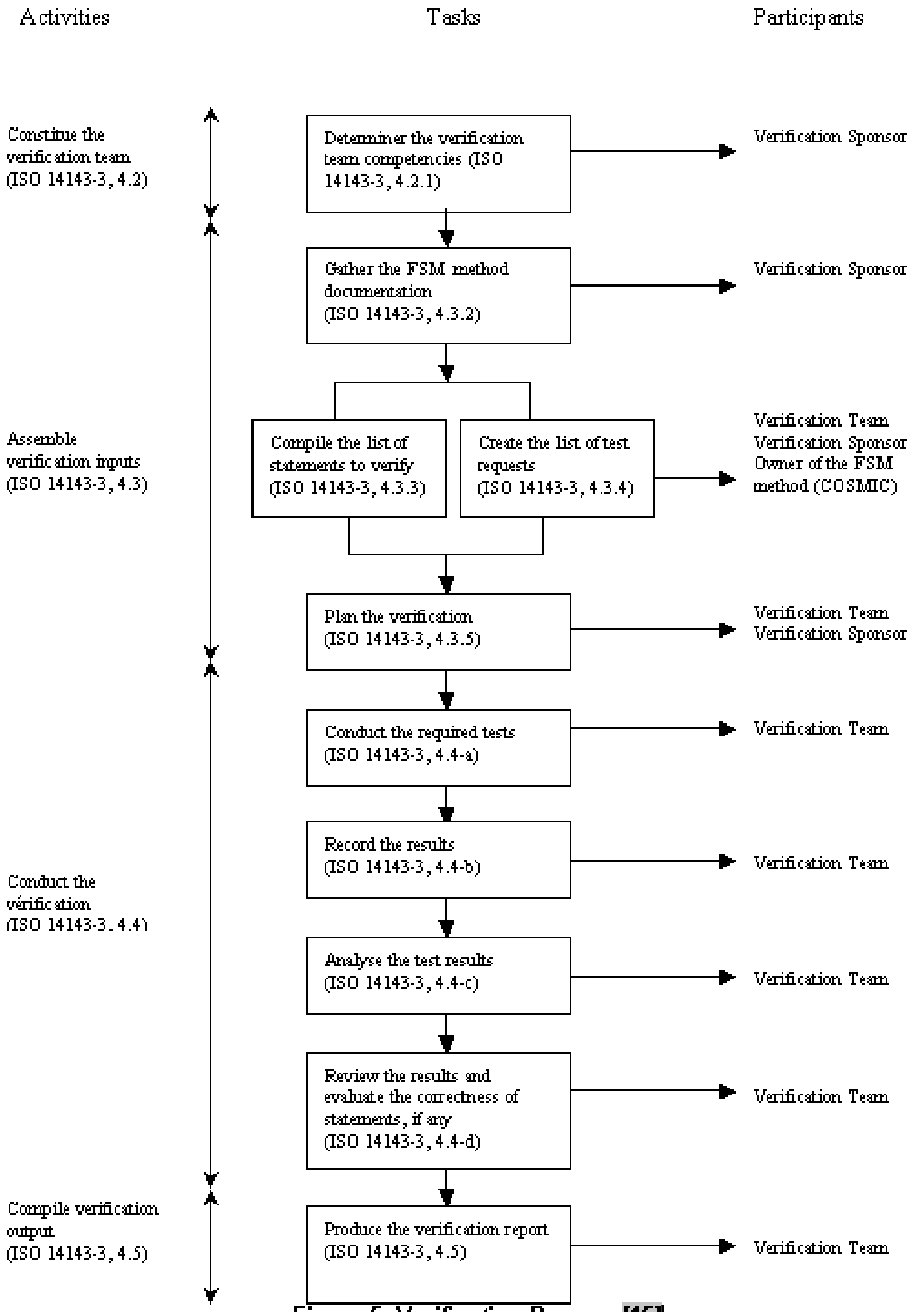

Figure 3: Verification Process [17]

Of course, it is expected that refinements will be required to the application of these principles and to our initial design of the System of References for software measurements.

This project will benefit from the direct collaboration of experts from industry. The objective is to define software measurement references and a common language for them.

The participation of some experts from the ISO/IEC JTC1/SC7/WG12 working group will provide greater credibility.

4.2 A catalog of measurement results

The measurement results coming from the application of the COSMIC-FFP method to the reference sets of FURs documented in ISO 14143-4 will be catalogued into a computer-based documentation system. Each of these ISO sets contains a number of RURs describing requirements of samples of management information system and real-time software. These RURs represent the inputs to the measurement process described in Figures 1 and 2. The catalog for this system of measurement references must include not only the measurement results, but also, for traceability purposes, the inputs to the measurement process. This context of ISO standardization adds relevance and legitimacy to this selection of case studies for our system of references, the purpose of such a system of references being to leverage and codify years of international experience.

This design, which includes both a verification process and a catalog, is illustrated in Figure 4.

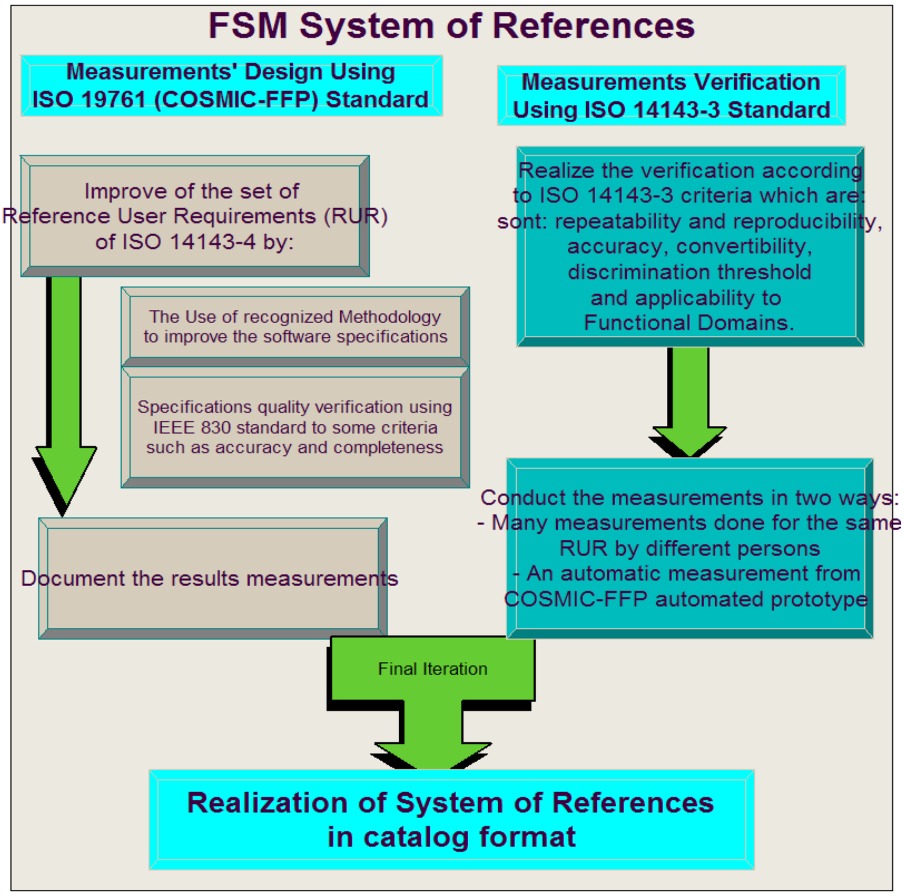

Figure 4: Experimental methodology for the FSM COSMIC-FFP System of References

4.3 Evolution of the system of references

Of course, the System of Reference will not be perfect initially, and will require an iterative cycle of improvements. Figure 5 below illustrates the iterative maintenance of the System of References.

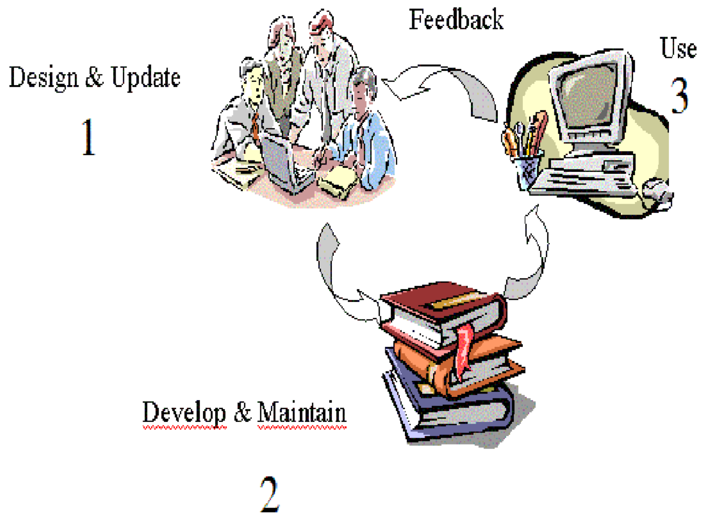

Figure 5: The System of References life cycle

5 Observations

Software Measurement is still emerging as a field of knowledge and, most often, traditional quality criteria of measurement methods, such as repeatability, reproducibility, accuracy and convertibility, have not even been investigated by software measurement method designers. In Software Engineering, the FSM community was the first to recognize the importance of such quality criteria for measurement, as illustrated in the recently adopted ISO document 14143-3; these criteria represent, however, only a subset of the metrology criteria which include, for instance, measurement units and internationally recognized measurement references (e.g. etalons). In this paper, we discussed the need, and an approach, for building a system of references using a specific FSM method, that is COSMIC-FFP (ISO 19761).

This proposal of a system of references for software FSM is a real challenge. Such a system of references has to consider not only the specifications of functional requirements, but also the experts' comments and the current practices in industry with respect to software size measurement. A system of references for software measurements will allow:

1. Researchers, users and experts to have independently verified values for reference;
2. Measurers to have confidence in the results of measurement;

3. The Software Engineering industry, to standardize the design of measurements and to verify the measurement tools.

In spite of the seemingly high cost of its construction and exploitation, this System of References could be quite beneficial if we compare it to the cost of project failures.

The content of this System of References for software FSM could be used next as input for ISO Software Engineering working groups tackling the standardization of measurement methods of various types, and not only for FSM. A research result by-product will therefore be a set of suggestions for improving the international standards themselves, thereby contributing to the maturation of the Software Engineering discipline.

The realization of a system of references could also contribute to the development of a theoretical analysis of measurement and to lay the foundations for a method of introducing into organizations, whether private or public, traceability of measurement results to international standards as codified in a system of software measurement references

References

1. Abran A. Analyse du processus de mesure des points de fonction. in *département du génie électrique et de génie informatique*, Thèse de Doctorat, Ecole Polytechnique de Montréal, Montréal, 1994, 405.
2. Abran A., Moore J. W., Bourque P., Dupuis R. and Trip L. *Guide to the Software Engineering Body of Knowledge*, Los Alamos, 2004.
3. Abran A. and Sellami A. Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology. in *IWSM*, (2002).
4. Abran A., Sellami A. and W. S. Metrology, measurement and metrics in software engineering. *IEEE Software Metrics Symposium, 2003. Proceedings. Ninth International*. 2-11.
5. Albrecht, A.J. Measuring Application Development Productivity. in *IBM Application Development Symposium*, (Monterey, CA, 1979), 14-17, 34-43.
6. Azzouz, S. and Abran, A. A proposed measurement role in the Rational Unified Process (RUP) and its implementation with ISO 19761: COSMIC FFP. in *Software Measurement European Forum - SMEF 2004*, (Rome, Italy, 2004).
7. Baize, D. and Girard, M.C. *Référentiel Pédologique*. Association française d'étude des sols, Paris, 1995.
8. BIPM, B.I.d.P.e.M.-. *Metre Convention and the MRA*, BIPM. http://www.bipm.org/en/convention/
9. Buglione Luigi and Abran Alain, The Software Measurement Body of Knowledge. in *Proceedings of 1st Software Measurement European Forum (SMEF2004)*, (Rome (Italy), 2004), 84-94.

10. Bureau National de la Métrologie. *Histoire de la mesure*, BNM- Bureau National du Métrologie. http://www.bnm.fr/mesure-metrologie/me.histoire1.htm

11. Fenton, N.E., Pfleeger, S.L. *Software Metrics : A Rigorous Approach*. Chapman & Hall, 1996.

12. Giles, A., E., Daich, G.,? T. *Metric Tools CROSSTALK*, February 1995. http://www.stsc.hill.af.mil/crosstalk/1995/02/Metrics.asp

13. IEEE Std 610.12-1990 *IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990*. The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1990.

14. ISO *Vocabulaire international des termes fondamentaux et généraux de métrologie*. International Organization for Standardization – ISO, Geneva, 1993.

15. ISO/IEC 14143-1 *Information technology – Software measurement – Functional size measurement – Definition of concepts*. International Organization for Standardization – ISO, Geneva, 1996.

16. ISO/IEC 14143-2 *Information Technology – Software Measurement – Functional Size Measurement – Part 2: Conformity Evaluation of Software Size Measurement Methods*. International Organization for Standardization – ISO, Geneva, 2001.

17. ISO/IEC 14143-3 *Information Technology – Software Engineering – Software Measurement - Functional Size Measurement – Part3: Verification of functional size measurement*. International Organization for Standardization – ISO, Geneva, 2002.

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

19. ISO/IEC 14143-5 *Information Technology – Software Measurement - Functional Size Measurement – Part 5: Determination of Functional Domains for use with Functional Size Measurement*. International Organization for Standardization – ISO, Geneva, 2004.

20. ISO/IEC 14143-6 *Information Technology – Software Engineering -- Software measurement -- Functional Size Measurement -- Part 6: Guide for use of ISO/IEC 14143 series and related international Standards*. International Organization for Standardization – ISO, Geneva, 2003.

21. ISO/IEC 15939 *Information Technology – Software engineering – Software measurement process*. International Organization for Standardization – ISO, Geneva, 2002.

22. ISO/IEC 19761 *Software Engineering – COSMIC-FFP – A functional size measurement method*. International Organization for Standardization – ISO, Geneva, 2002.

23. ISO/IEC IS 20926 *Software Engineering – IFPUG 4.1 Unadjusted functional size measurement method* - *Counting Practices Manual*. International Organization for Standardization, Geneva, 2003.

24. ISO/IEC IS 20968 *Software Engineering – MkII Function Point Analysis* - *Counting Practices Manual*. International Organization for Standardization, Geneva, 2002.

25. ISO/IEC IS 24570 *Software Engineering – NESMA functional size measurement method version 2.1*. International Organization of Standardization, Geneva, 2004.

26. Jacquet J.P., Abran, A. Metrics Validation Proposals: A Structured Analysis. in *IWSM*, (1998).

27. Jones, C. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw Hill, New York, 1996.

28. Paton, K. and Abran, A. *A Formal Notation for the Rules of Function Point Analysis*, Université du Québec à Montréal, Montreal, April 28, 1995.

29. Rubey, R.J. and Hartwick, R.D. Quantitative Measurement Program Quality. in *National Computer Conference*, (1968), 671-677.

30. Software Measurement Services. *The Power of COSMIC FFP*, Software Measurement Services. http://www.gifpa.co.uk/news/8/p2_cosmic.html

31. Zuse, H. *A Framework for Software Measurement*. DeGruyter Publisher, Berlin, 1998.