The contribution of metrology concepts to understanding and clarifying a proposed framework for software measurement validation

Asma Sellami, Alain Abran
École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest, H3C 1K3 Montréal Québec, Canada,
`asma.sellami.a@ens.etsmtl.ca` `aabran@ele.etsmtl.ca`

Abstract: *In the software engineering literature, numerous practitioners and researchers have proposed hundreds of "software measures", or "software metrics". To help industry assess the quality of these proposed measures, various researchers have proposed various approaches to software measurement validation, none of which has yet been widely used by either designers or users of software measures. To tackle this diversity of validation approaches, Kitchenham et al. had proposed a framework for software measurement validation and suggested a critical review of their proposed framework. This paper performs such a review using a key ISO document on measurement, that is the ISO Vocabulary on Metrology as well as a measurement process model derived from an analysis of the individual validation proposals. The metrology concepts in particular have facilitated greater understanding of the set of measurement sub-concepts that must be included in each of the steps from the design of a measurement method to the use of the measurement results.*

1   Introduction

In the software engineering literature, hundreds of metrics have been proposed to address various topics of interest to the industry, such as the measurement of software complexity or of object-oriented elements [11, 15 17, 18, 19, 34-36]. It is to be observed, however, that many of these software metrics have been based on an "intuitive" approach. For instance, many software metrics are often described only in the form of an algorithm, and are based on terms like "process", "flows of data" without a precise definition for each of the terms within the proposed algorithm. Software metrics designers most often do not formally specify the meta-model for the proposed metrics, do not refer to formal agreed-upon definitions of terms, and do not empirically verify quality criteria such as accuracy, repetitiveness and repeatability [22]. This lack of exactness explains why the results given by these measures are context-dependent, i.e. dependent on the way the users of these measures understand the terms. Therefore, a significant number of so-called software metrics are not based upon verifiable foundations and would not qualify as measurement methods.

Furthermore, even for the IFPUG Functional Size Measurement method[20], there is still, after twenty years, ambiguity about what a Function Point represents, and there remain questions about its validity with respect to

measurement theory [1-3, 8, 9, 14, 16]. For [22], even though the characteristics and the numerical assignment rules are defined in greater detail in [20], the meta-model of the concept to be measured has not yet been clarified. Consequently, the mathematical links between this concept and the numerical assignment rules are not yet fully established (this does not mean, of course, that they cannot be established). This implies that no one has yet documented, in a formal way, that this measurement method is consistent and that it measures what it is supposed to measure. Nevertheless, there is evidence of its usefulness in prediction systems with data from the Management Information System (MIS) domain in the 80s and 90s. This would mean that its validity as a measurement method has not been fully demonstrated even within its original domain of applicability (e.g. MIS in the 1980s-90s timeframe), and limited documented support for generalization to other domains (ISO 14143-5 [26]), including types of MIS software more typical of the 2000s.

In software engineering, various authors have investigated the issue of software measurement validation, also often referred to as "software metrics validation" by some authors [10, 13, 31, 32]. What is a valid software measurement (or metrics), and how do you validate it? Various authors have attempted to address these questions, but usually from differing points of view (mathematical, practical, etc.) in the past few years. For example, Schneidewind proposes a metrics validation methodology based on six validity criteria [32], while other authors postulate that a measurement method is valid if it can be shown that it gives a proper numerical characterization of some attributes [13, 17]. Even so, this definition is far from being unanimously accepted, and Fenton remarks that, in the software community, it is expected that validation must also entail demonstration that the measure is itself part of a valid prediction system [13].

Kitchenham et al. note that "what has been missing so far is a proper discussion of relationships among the different approaches" [10]. They recognize the need for a validation framework which would take into account, and integrate, the distinct validation perspectives proposed by these various authors, and they put forward a proposal for such a validation framework, encouraging researchers and practitioners to respond critically to it.

The goal of their initial framework was to help researchers and practitioners alike to understand how to validate a measure, how to assess validation methods and when it is appropriate to apply a measure in a situation requiring several validation criteria.

The software engineering literature has proposed multiple viewpoints for addressing software measurement validation, such as are represented by the following questions:

- Is the measure internally valid, i.e. can it be shown that it gives a proper numerical characterization of the attribute to be measured?
- Is the measure usable? A measure as perfect as possible from a mathematical viewpoint would not be of any interest if it were not possible to apply it (far too time-consuming, for example).
- Can the measure take its place in a valid prediction system?

The measurement validation framework proposed in [10] has the following three headings:

- Structure of measurement
- Models and measurement
- Software measurement validation

A number of concepts are presented under each heading, as listed in Table 1.

Table 1: Structure of the validation framework presented in [10]

Headings	Measurement concepts discussed under each heading
Structure of measurement	Entities, attributes and their relationships
	Units and scale types and their relationships
	Values
	Properties of values
	Measurement instrument
	Indirect measures
	Compound units
	Properties of indirect measures
	Implications for measurement validation
Models and measurement	Unit definition models
	Instrumentation models
	Attribute relationship models
	Practical problems with attribute models
	Measurement protocols
	Differences among the definition models
	Entity population models
Software measurement validation	Theoretical and empirical validation
	Theoretical measurement validation issues
	Corroborating measure empirically
	Validating empirical relationships between attributes

The measurement validation framework proposed in [10] attempts to answer all of the above. However, it is challenging to understand and use; for instance, the expression measurement validation is used in many ways with different

meanings, the authors not stating explicitly which specific sub-concept of measurement is being addressed.

This paper reviews the validation framework proposed in [10], and uses recent work on the modeling of ISO metrology concepts [5, 6] and a broader model of measurement process steps [4, 28, 29] to assess it, to highlight its coverage and, finally, to propose some clarifications and further refinements.

This paper is organized in the following way. Sections 2 and 3 present the two sets of measurement concepts used to analyze the framework proposed in [10]: a model of the metrology body of knowledge [5] and a broader model for software measurement methods [28]. Section 4 describes and then positions the different elements of Kitchenham's validation framework with respect to these metrology and measurement concepts. The concepts are discussed in sections 5 through 8. Section 9 presents a summary of this analysis.

2   Metrology and measurement process models

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, sometimes built up over centuries, commonly referred to as "metrology". This metrology domain is supported by governmental metrological agencies, which are to be found in most industrially advanced countries.

The ISO document that represents the official international and legal consensus in this field is the ISO Vocabulary of basic and general terms used in metrology [21]. While this ISO document is widely known in the metrology field, it is almost unknown in the "software metrics" community, not being quoted by any of the authors who have addressed the topic of software measurement validation.

The ISO Vocabulary follows some of the concepts underlying 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 our understanding, of this complex set of interrelated concepts, an initial set of models for the various levels of metrology concepts within the ISO Vocabulary was presented in [5].

The high-level model of the set of categories of metrology 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, i.e. 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 "measurement" in the sense of measurement operations, while the control variables are the "étalons"[1] (official yardsticks) and the "quantities and units". It is to be noted that the measurement operations, and, of course, the measurement results, are influenced by the "characteristics" of the measuring instruments. This combined topology of concepts corresponds then to the concept of a "measuring instrument" in Figure 1.

In the ISO Metrology Vocabulary, the term "measurements" used as a single term corresponds to the "set of operations" used for measuring. Also, in all figures and tables in this paper, a term taken directly from the ISO Vocabulary will appear in roman type, while terms representing concepts not specifically listed will appear in italics; for instance, in Figure 1, we have added the term "Input", which is not included in any of the six categories of the ISO Vocabulary. More detailed models of each of these six categories of metrology terms are presented in [5], three of which are presented in Figure 2 and Figure 3, as well in Table 2.
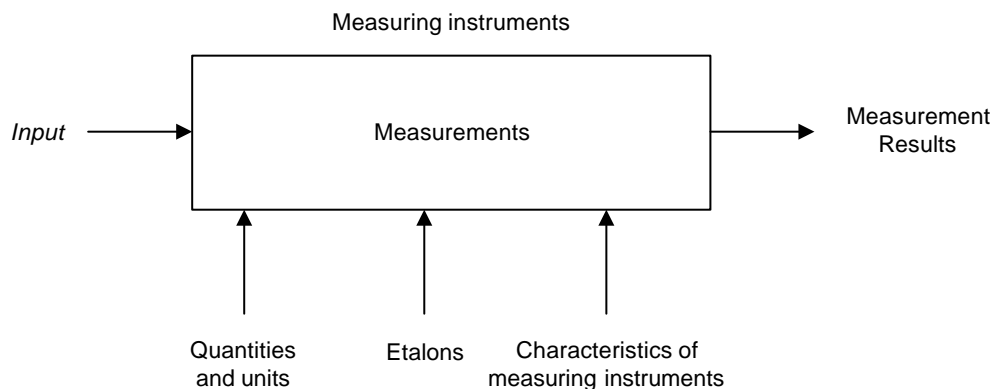
**Figure 1:** Model of the categories of metrology terms (Source: [5])

The term "metrology" itself 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.

---

[1] Étalon: an example of an internationally recognized material yardstick: the physical "meter" etalon in length measurement recognized as the official "étalon" for the meter. Etalons are also refined over time. For instance, the official definition of the meter changed in 1983 to the distance traveled by light, in an empty medium, in 1/299,792,458[th] of a second.

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 and an output representing the "results of measurement", plus, of course, various control variables.

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 impacted by an "influence quantity" during the measurement 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 2. Again, this structure is our own.
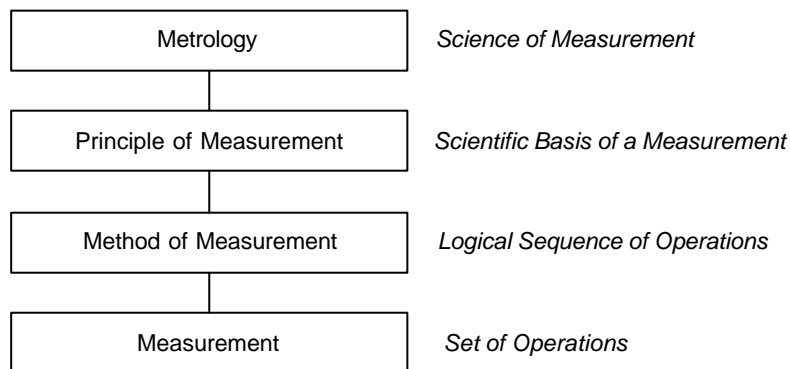
Figure 2: Measurement foundations (Source: [5])
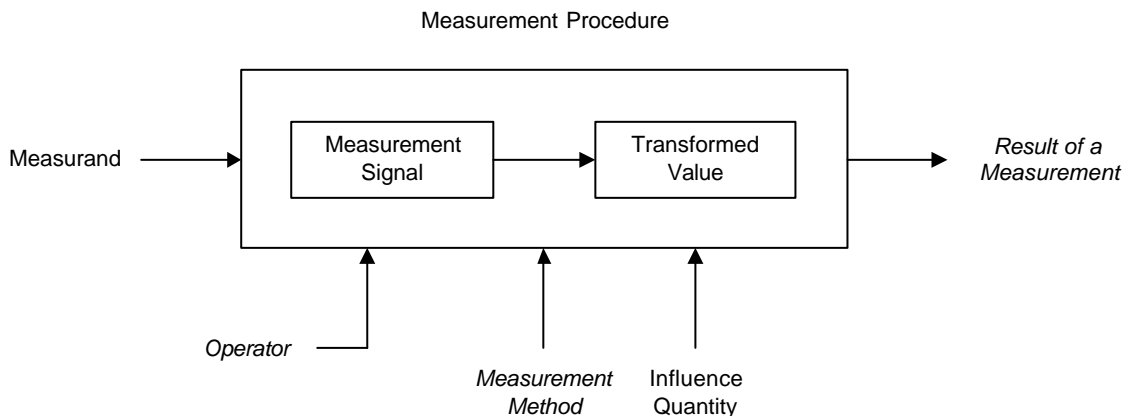
Figure 3: Measurement procedure (Source: [5])

Table 2: Classification of terms in the "Measurement Results" category
(Source: [5])

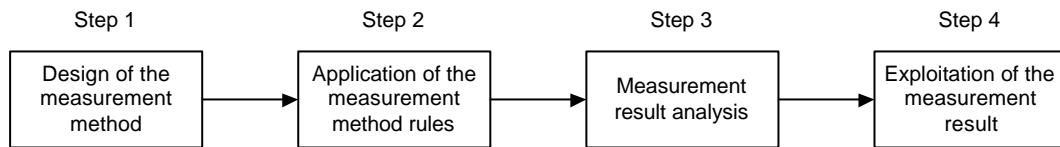| *Types of measurement results*	*Modes of verification of measurement results*	**Uncertainty of measurement**
Indication (of a measuring instrument) Uncorrected result Corrected result	Accuracy of measurement Repeatability (of results of measurements) Reproducibility (of results of measurements)	Experimental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

3   A Broader Measurement Process Model

In their work as ISO editors for the Guide to the Verification of Functional Size Measurement Methods (ISO 14143-3)2 [22], Abran and Jacquet studied the work of the various software engineering authors who had dealt with "metrics validation" [4, 28, 29]. Significant variations were found in the authors' approaches, as well as in their use of similar terms, but very significant differences in the related concepts. To clarify the confusion caused by the inconsistent terminology used by these authors, a broader measurement process model was proposed (Figure 4) identifying four distinct steps, from the design of a measurement method to the exploitation of the measurement results [28]. Then, the approaches of the various authors, as well as the validation concepts that were being addressed differently by these authors, were sorted, depending on whether or not they were addressing validation issues related to Steps 1 to 4 of the process model in Figure 4 [28].
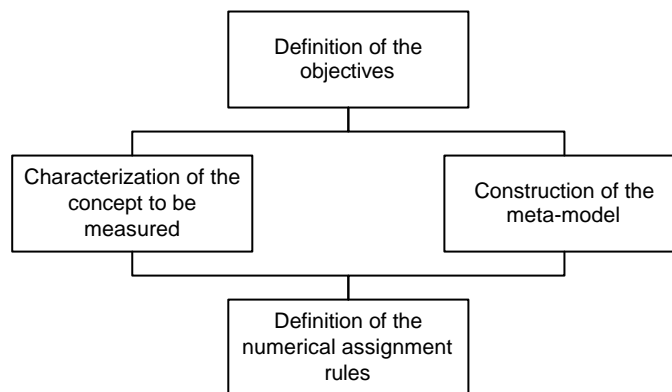
Step 1 is further broken down in Figure 5, where the sub-steps required for the design of a measurement method are decomposed. An example of the instantiation of this model of the design of measurement method is provided by the COSMIC-FFP Functional Size Measurement method [7], as documented in ISO 19761:2003 [23]. The objective is to measure software functionality as defined in ISO 14143-1: 1998 [25]. The attributes to be measured are specified as the data movements of the functional processes, and the meta-model is composed of four data movement types based on the generic COSMIC software model. The numerical assignment rule is then specifically stated through a measurement function where one COSMIC functional size unit (i.e. 1 Cfsu)

---

[2] The initial target for this ISO document was a Validation Guide, but, due to the lack of consensus on software measurement validation, the scope of the ISO document was limited to a Verification Guide for some of the properties of measurement methods as described in the ISO Metrology Vocabulary.

corresponds to a unit of 1 for any type of data movement (of course, each of these terms and concepts is precisely defined within both the ISO COSMIC-FFP glossary and the measurement rules [23]).

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

Figure 4: Measurement Process – High-level Model (Source: [4, 28, 29])

Figure 5: Detailed model of Step 1 – Design of the measurement method (Source: [4, 28, 29])

In the literature, in software engineering as well as in psychology or physics, the words (nouns) "measure" and "measurement" are used in different ways:

- to refer to a method allowing the assignment of a numerical (or symbolic) value to an object in order to characterize one attribute of this object;
- to refer to the application of this method;
- to refer to the result of this application;
- to refer to the process from the design of a measurement method to its exploitation.

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

It is to be noted that very few of the measurement concepts in the ISO Metrology Vocabulary address the first step (design of a measurement method) and none address the last step (exploitation of the measurements results) of [28]. This is illustrated in Table 3, which depicts an initial mapping between Figure 1 and Figure 4.

Table 1: Alignment of ISO metrology concepts with the measurement process model (Source: [6])

Measurement process model [4, 28, 29]	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 Categories of Metrology Terms [5]	Quantities and units	Measuring instruments, and their characteristics	Measurement results	

As illustrated in Table 3, the measurement process model derived in [28] from the validation proposals has a broader scope than the ISO Metrology Vocabulary model: for instance, it includes more concepts in the first step than simply "quantities and units" (see Figure 5). In summary, the ISO Metrology Vocabulary model [5] does not handle either the full design of the measurement method or the exploitation of the measurement results; it does, however, discuss in greater depth the measuring instruments and their characteristics. Thus, these two models are complementary and are used here as analytical tools for reviewing and proposing improvements to the framework in [10].

It was noted in [28] that the question of how to validate a measure, stated in this way, had not been precise enough and led to multiple interpretations, and corresponding dissimilar validation proposals dealing with the distinct measurement-related concepts being investigated.

There is, therefore, a need for a revised validation framework which is broader in scope, more precise and able to consolidate these various measurement views. Such a broader validation framework can be proposed on the basis of the various steps of the metrology and measurement process and should clearly state the four distinct measurement-related concepts that should be validated (Table 4), and validated using distinct criteria, since each is clearly distinct, that is, each represents distinct entity type: the design of a measurement method, the design of the related measurement instrument, the measurement results themselves, and the use of the measurement results in a specific context for a specific purpose.

Table 4: Distinct measurement entity types subject to validation criteria

Step1 Validation of design of the measurement method	Step2 Validation of the measuring instrument	Step3 Validation of the measurement results	Step4 Validation of the exploitation of the measurement results

4 The validation framework proposed in [10] and corresponding entity types

To analyze the framework proposed in [10] using the distinct measurement entity types subject to validation, as listed in Table 4, it is necessary to first analyze the list of measurement terms discussed in [10] and map them to the sets of concepts from Table 3. The results of this mapping are presented in Table 5. It can then be observed that, under each of the three section headings in [10], there is a discussion of measurement concepts spanning the full range of measurement-related concepts, with the "structure of measurement" dealing mostly, but not exclusively, with the design of measurements methods, the "models and measurement" section dealing mostly with measuring instruments, and the third section on "software measurement validation", dealing with entities and criteria, exclusive of the design of measurement methods.

Table 5: Mapping of the validation framework in [10] to the entity types subject to validation

Kitchenham et al. Validation Framework headings	Kitchenham et al. measurement concepts discussed under each heading	Step 1 Design of the measurement method (including quantities and units)	Step 2 Application of the measurement method rules (including measuring instruments and their characteristics)	Step 3 Analysis of the measurement results	Step 4 Exploitation of the measurement results
Structure of measurement	Entities, attributes and their relationships	×			
	Units and scale types and their relationships	×			
	Values	×		×	
	Properties of values	×			
	Measurement instrument		×		
	Indirect measures				×
	Compound units				×
	Properties of indirect measures				×
	Implications for measurement validation	×		×	
Models and Measurement	Unit definition models		×		
	Instrumentation models		×		
	Attribute relationship models			×	
	Practical problems with attribute models				×
	Measurement protocols		×		
	Differences among the definition models		×		
	Entity population models				×
Software Measurement Validation	Theoretical and empirical validation		×		
	Theoretical measurement validation issues			×	
	Corroborating measures empirically				×
	Validating empirical relationships between attributes				×

It can be observed that, in [10], the terms "validation, and "measurement validation" are always used in the broad sense, without explicitly stating which of the detailed measurement concepts, or sub-concepts, are being investigated and discussed.

It should be noted also that in [10], the expression "empirical relations system" is used in two different contexts, each with its distinct meaning and requirements:

1.   The first context is the empirical relations system between the real world and the mathematical world for a single attribute at a time.

2.   The second context is the empirical relations system among several attributes.

Each will be discussed later in the selected metrology context.

The next sections present an analysis of [10] based on the structure of a process model for software engineering measurement methods [5, 28], identifying the four distinct steps involved, from the design of a measurement method to the use of the measurement results.
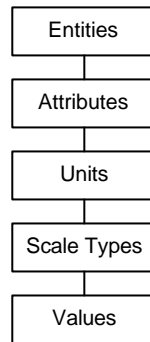
5   Design of a measurement method: elements to be validated

5.1   The measurement entities as objects of validation in the design

When entities, attributes, units and the representation condition are discussed in [10], it is to deal explicitly with the design of a "measure", that is, with the "measurement method" designed to capture, in a numerical representation, the attribute of the entity of interest, whether an object or a concept. Indeed, for measuring, it is necessary to use the design of a measurement method, if one exists, or to design one if it does not. It was seen in [28] that to build a measurement method, it is necessary to define the objectives, to select the meta-model of the concept or object to be measured, characterize the attribute(s) to be measured and to define the numerical assignment rules, including quantities and units (Figure 5).

How does the validation framework presented in [10] deal with the design of a measurement method, that is, does it completely cover the detailed set of sub-steps identified in [5, 28]?

Figure 6 presents our hierarchical model of the measurement concepts mentioned in [10] as they relate to the design step of the measurement method, that is: entities, attributes, units, scale types and values.

**Figure 6:** A hierarchical model of a measurement method design concepts, as discussed in [10]

It can be observed when Figure 6 is mapped into Figure 5 that the concepts related to the entities and attributes to be measured are specifically stated while the numerical assignment rules are partially addressed in terms of units, scale types and values. However, the reference in [10] to the empirical relations system between the real world and the mathematical world for a single attribute is not precise enough and does not specify the concepts required for validation of the design of a measurement method, that is, the objectives of the measurement method, the characterization of the concepts (through the attributes), the meta-model of these attributes derived to meet the objectives of the measurement method or the rules for the numerical assignments to units and values.

5.2   Attributes combination as an object to be validated

A second model is proposed in [10] to represent a phenomenon described by a combination of attributes, which is referred to as an "indirect measure" in that work. For example, the concept of speed is defined by a combination of two other measurements (i.e. distance and time), and represented by the ratio of distance covered per unit of time, or distance/time. It should also be noted that multidimensional attributes (such as in physics) can be represented (i.e. defined) by vectors (e.g. velocity which is defined as a combination of speed and time) or scalars (e.g. speed which is measured in terms of distance per unit of time).

The examples in [10] are referenced as representations of indirect measurements, and it is stated there that such indirect measures are to be represented by an equation defining these "measurements" based on an empirical association between attributes formalized as a mathematical equation (the example given is the program size used in an equation to predict project effort). However, a closer look at the text in [10] reveals that the elements presented as criteria for validation do not deal with the concepts illustrated in the example provided, but rather with other concepts which are referred to as the

definition of ratios, indicators or formulas to describe a combination of attributes for an object of interest. They do not "explain" a relationship across attributes of distinct entities. In Figure 7, we propose a model of the concept of "association of attributes" on the basis of the descriptive text in [10] (instead of Figures 2 and 3 in [10] :
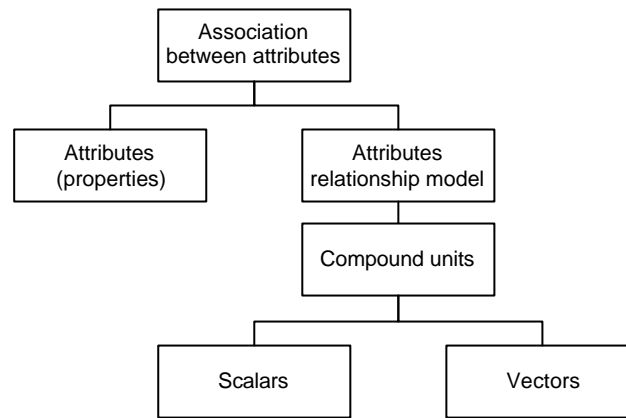
Figure 7: A model of the "association between attributes" on the basis of the text in [10]

The ISO model of software quality described in ISO 9126-1: 2002 would be a representative example of an instantiation of an association between attributes [24], as illustrated in Figure 7. This ISO model of quality in 9126 represents software quality through six quality characteristics, each characteristic being decomposed further into sub-characteristics, each sub-characteristic being described in numerical terms through a proposed selection of measures described through either a single number or a combination of ratios and indicators (which in turn could be represented by scalars or vectors).

It is to be observed that ISO 9126 proposes an open model[3] of attribute relationships in the sense that even though it normalizes its meta-model and detailed descriptions, it is not prescriptive in the sense that it does not prescribe a specific selection of characteristics, nor a specific set of numerical assignments within the prescribed relationship model.

5.3   One of the empirical relations systems: the relationship between an attribute and its measurement

Attributes are the properties that an entity possesses, and a "measure", by definition, transforms attributes into real numbers. Thus, for a given attribute, there is a relationship in the empirical world which we want to capture formally in the mathematical world [10]. The example given in [10] is that of two people,

---

[3] To be more precise, ISO 9126 specifies three 'open' models of quality for software products: internal quality, external quality and quality in use.

one of whom is taller than the other. A "measure" allows us to capture the "is taller than" relationship, to map it to a formal system and to explore the relationship mathematically. It is to be noted that, even though this example is based on two references (two distinct people), this recourse to explicit measurement references is not formally stated in the framework proposed in [10], nor is there a recognition that such a reference should be fixed, and not vary across an instantiation of measurements. Furthermore, while the example refers to numbers, these numbers are of the "ordinal scale type", however there is no reference to a "scale" standard.

An improved measurement validation framework should therefore include additional metrology-based criteria, such as measurement references for the representation conditions, including references related to measurement standards (such as 'etalons' in the metrology domain). Of course, since software measurement methods have not yet reached maturity and recognition in the international standardization community, we suggest that well-documented case studies be temporarily recognized as substitute references.

Table 6 presents some of the analytical tools to be used for validation of the design of a measurement method: software modeling rules for the meta-model, numerical assignment rules for the mathematical representation and either 'étalons' or case studies for the measurement references.

Table 6: Measurement method design - examples of validation criteria and tools

Design of measurement method		
Meta-model	Measurement References	Mathematical Representation
Software modeling rules	Etalons Case studies	Numerical assignment rules

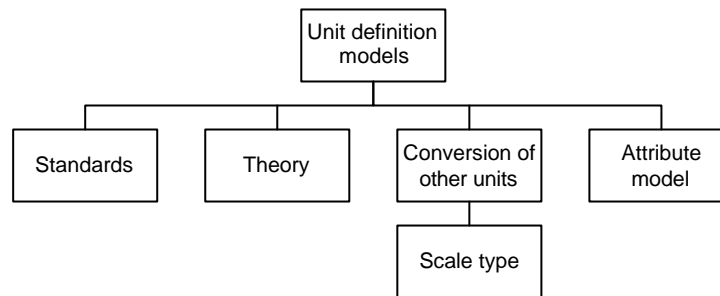## 6   Validation of the application of a measurement method

Step 2 in Figure 4 refers to the application of a measurement method in a specific instantiation. Validation within this step requires the validation of both the measuring instruments and their characteristics at the time of the instantiation.

6.1   Concepts within [10]

The type of validation referred to above is partially addressed in [10] in the second part of the framework labeled as "models and measurement", which includes the following set of terms from [10]: unit definition models, instrumentation models, attribute relationship models, practical problems with

attribute models, measurement protocols, differences among definition models
and entity population models.

Figure 8 and Figure 9 present our modeling of the major sets of concepts
presented textually in the second part of the proposed framework from [10]:
"unit definition models" and "measurement process" as labeled in [10].

Figure 8: A model of "unit definition models" on the basis of the text in [10]
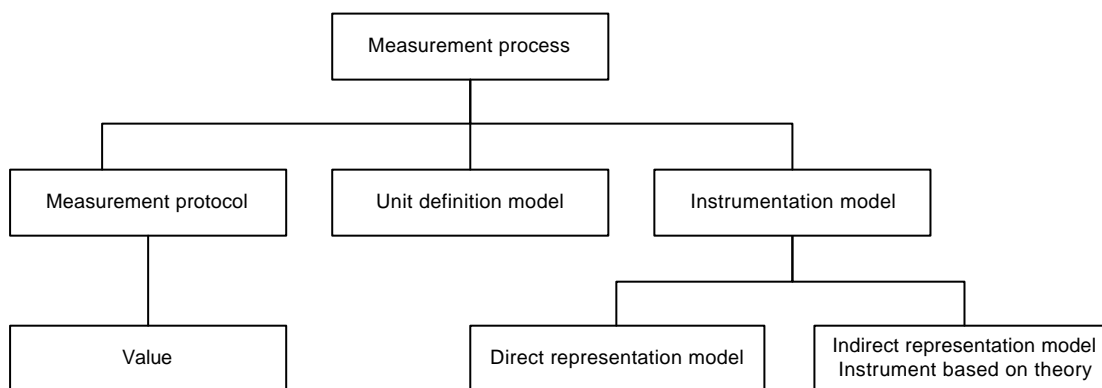
Figure 9: A model of the "measurement process" on the basis of the text in [10]

In our opinion, the "unit definition models" in Figure 8 would belong to the
design of a measurement method, rather than to a specific instantiation, while
the "measurement process" (described in [10] and represented here in Figure 9)
is a different view of the ISO "measurement procedure" as modeled in Figure 3.

Thus, the validation of the application of a measurement method must refer to
the validation of the measuring instrument and the characteristics of this
measuring instrument in a specific instantiation.

6.2   Validation of the measuring instrument

We think that, if a measuring instrument is valid, the characteristics of this
measuring instrument would be valid in a given environment. The validation of
the measuring instrument characteristics should deal with quantitative and
qualitative characteristics, as well as the functionality test in the use and control

of the instrument, and the measuring range (see Table 4 in [5] for further details). It is also necessary to verify that the measurement method behind the measuring instrument is valid too.

6.3 Validation of the measuring instrument characteristics

This type of validation is required to determine whether or not the characteristics of the measuring instrument are valid. It is also essential to ascertain whether or not the software modeling rules are well defined enough for the instrument to measure what it purports to measure. This is modeled in Figure 10 where the double arrows indicate a reciprocal influence between two elements.
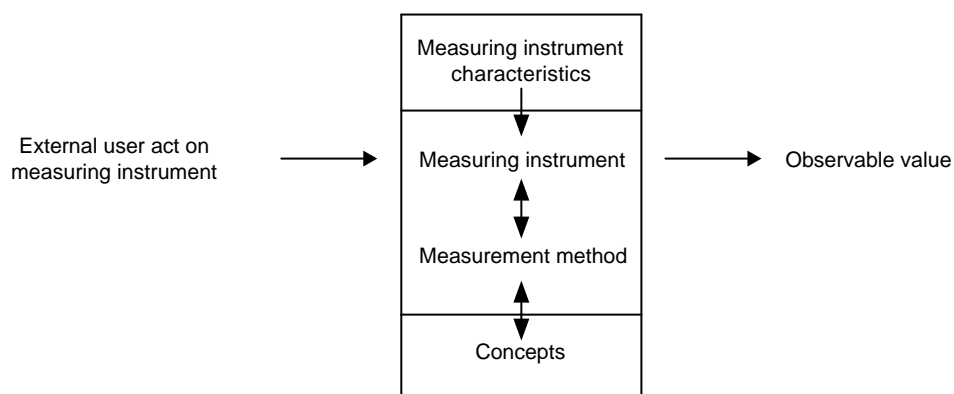
Figure 10: Validation elements of measuring instrument

In this table, there cannot be a valid measuring instrument without a valid measurement method or model associated with it already in place. Also in this table,, the simple arrows indicate a causal influence in one direction only.

7   Elements of validation for the measurement results themselves

Following the application of measurement method rules, the results obtained can be exploited as a source of knowledge according to the analytical world defined, and agreed upon, by the measurement method used. This topic is addressed within every section of the document [10] to identify the need for validation of the results of measurement methods. According to the mapping in Table 5, elements of this step are addressed in the following sections of [10]: the structure of measurement, in order to obtain an interpreted value; models and measurement, by considering two types of attribute models, definition or predictive; and software measurement validation, to addresses some criteria about theoretical measurement validation issues.

8   Validation of the exploitation of measurement results

8.1   Models of exploitation of measurement results

Once a measurement result is obtained, it can be exploited in a quantitative or qualitative model for a specific intended use, such as a quality model, an assessment model, a descriptive model of relationships, a prediction model, etc. In [10], this is referred to as a "compound attributes" model, and includes various concepts which we have modeled in Figure11 on the basis of the text of these authors.
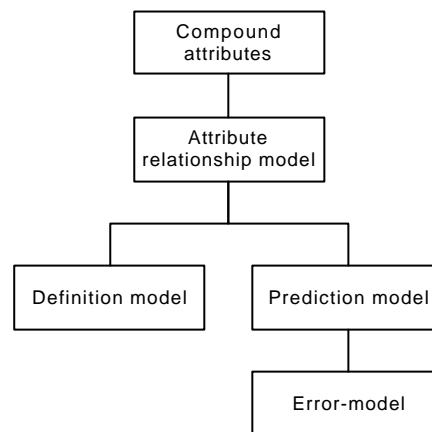
Figure 11: A model of "compound attributes" on the basis of the text in [10]

In software engineering, a significant number of authors on software measurement have included the validation of these models either as part of measurement validation (such as in [10]) or, like Schneidewind [32, 32], as the main topic of measurement validation. This is, however, a rather unusual view of measurement validation, since most of the related concepts discussed by these authors do not refer to the detailed field of measurement, but rather to the field of experimental studies, as well as to the field of data analysis tools (such as statistical models, etc.).

It should be noted in particular that the requirements of validation are much more constraining than the requirements of verification. For example, we can verify the performance, or "quality", of an estimation model with a number of criteria, such as correlation coefficient, determination coefficient (R2), coefficient of prediction - Pred (25%) < in 75% of the cases. A verification process provides quantitative values for evaluating a model, in a specific context. However, these values do not "validate" a model; for example, a model of regression is valid according to its conditions of construction and interpretation, and not according to the values found in a particular context.

8.2   Relationships among multiple attributes

According to [10], verification of the empirical relationships among attributes requires to:

- Corroboration if the relationship is an association or if it is causal. Observation of a statistically significant correlation is not sufficient to corroborate a causal relationship. To confirm causality, it is necessary to control the attribute, which generates the relationship (the independent variable) and to confirm that the expected change is observed in the other attribute (the dependent variable). However, it is often difficult to control software attributes. In this context, the analysis in [10]is not based on the field of metrology, but rather on the field of experimental studies (and related methodologies) in general.
- Consideration of whether or not the method of investigating a relationship is likely to change the relationship itself. For example, to verify that a cost model produces accurate predictions, it is appropriate to compare predicted effort and duration with actual effort and duration.
- Consideration of the difference between goodness of fit and predictive capability.

It is stated in [10] that, to establish whether or not a "measure" can be used to predict the value of another "measure", we need to establish whether or not there is a relationship between the two attributes being measured. Similarly, when Schneidewind suggests in his paper "Methodology for Validating Software Metrics" [32] that the control of the relationships between attributes should be a fundamental method for "software metrics validation", our analysis of his text indicates that this author does not discuss measurement design validation for the numerical representation of an attribute, but rather the use of these measured attributes in certain contexts (e.g. in models of use, such as evaluation, control and the prediction of other attributes).

9   Summary of observations and recommendations

The objective of the validation framework proposed in [10] was to propose a basis for a consensus on "measurement validation", because this is considered critical for the success of software measurement. However, we have seen that, in the specialized field of measurement, the expression "measurement validation" is rather broad and can lead to confusion.

For example, Schneidewind [32] proposed a validation process based mainly on the analysis of the results of measurement methods (Step 3 of Figure 4: Measurement Results Analysis). Fenton [12, 13], for his part, suggested that, in order to be valid, a measurement method must satisfy the representation

condition of measurement theory. In other words, he proposed a validation process which addresses only the relationships between the two substeps "characterization of the concept to be measured" and "definition of the numerical assignment rules" of the first step in Figure 4.

By contrast, Kitchenham et al. [10] proposed a validation process addressing some elements and criteria from all parts of the measurement method process as described in Figure 4. Nevertheless, the analysis reported here on this validation framework (and summarized in Table 7) illustrates that it does not cover the full spectrum of the process model of measurement methods. For example, it does not identify the need for the validation of the meta-model and its relationships with the different components of measurement methods.

This paper has illustrated the need for a more precise use of measurement terminology in order to avoid ambiguity, clarify the entity types being validated and understand the corresponding properties that must be addressed in a measurement validation framework. For instance, where in [10] the terms "measure" and "measurement" are used to talk about the generic object of validation, we have illustrated the usefulness of the more precise vocabulary derived from the metrology domain of knowledge.

Table 7: Classification of validation elements of the two software measurement method

Metrology and measurement process model	Kitchenhams' framework
Step1: Design of measurement method	
Definition of objectives	Indirectly: Capture the attribute of the entity (concept, object) in a numerical representation
Design or selection of meta-model	Not specified
Characterization of concept to be measured	Identify the elements of measurement and their properties (entities and attributes) Identify how to define those elements (attribute validity) Combine attributes
Definition of numerical assignment rules (including quantities and units)	Define appropriate theoretical and empirical methods of those properties Representation condition should be satisfied and numerical assignment rules should properly characterize the attribute measured (Units, scale types, values)

Step2: Application of measurement method / Design of measuring instrument	
Software documentation gathering	Not specified
Construction of the software model (including characteristics of the measuring instrument)	Structure model of software measurement Properties of individual elements Theoretical validation
Application of the numerical assignment rules	Not specified
Step3: Measurement result analysis	
Measurement result	Models used to define the elements in the structure model When to create / apply a measure Properties of measurement elements Empirical validation
Step4: Exploitation of measurement result	
Use of measurement result	Issues involved in validation from the viewpoint of the properties of measurement elements Predictive system Entity population models

Recourse to this specialized metrology vocabulary has helped to highlight, for instance, that both the design of a measurement method, and the use of the measurement results had been discussed, but with different criteria. Similarly, it was highlighted that in [10] the expression 'empirical set of relations' refers not to one but to two distinct concepts: the representation condition of a single attribute at a time and relationships across multiple concurrent attributes, and sometimes even about objects of a distinct nature (such as size and effort in a causal model with dependent and independent variables).

Using both the metrology vocabulary and a process model of measurement methods, we have illustrated that, even in the validation framework proposed in [10], several facets of the measurement process in [4], without making distinctions as to which facets (or sub-concepts,) are treated and several are not. Thus, we note in these papers the need to make a distinction between measurement concepts and sub-concepts (i.e. steps in [28]) to which to apply the proposed validation criteria.

This immaturity in software measurement has been noted by Kitchenham: "A practical (validation) framework is an ambitious goal that requires input from practitioners and the research community" [10]. The impact of this on software

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

Finally, for future research work, we suggest that a consolidated and improved framework be built on the basis of the three measurement models discussed in our analysis: ISO [5], Abran et al. [28], and Kitchenham et al. [10]. In summary, this means, then, that not one, but multiple validation types are required, depending on the types of entities to be validated, and each with its own relevant set of validation criteria.

References

[1]   A. Abran, "Analyse du Processus de Mesure des Points de Fonction", in Thèse de doctorat de l'université de Montréal. Montréal, 1994.

[3]   A. Abran and P.N. Robillard, "Function Points: A Study of Their Measurement Processes and Scale Transformations", Journal of Systems and Software, vol. 25, no. 2, 1994, pp. 171.

[4]   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.

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

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

[7]   A. Abran, J.M. Desharnais, S. Oligny, D. St-Pierre, and C. Symons, "COSMIC FFP - Measurement Manual, Version 2,1", Montreal 2001.

[8]   B. Kitchenham, "The Problem with Function Points", IEEE Software, vol. 14, no. 2, 1997, pp. 29-31.

[9]   B. Kitchenham and K. Kansala, "Inter-Item Correlations among Function Points", IEEE Software Metrics Symposium, 1993, pp. 11-14.

[10]   B. Kitchenham, S.L. Pfleeger and N. Fenton, "Towards a Framework for Software Measurement Validation", IEEE Transactions on Software Engineering, vol. 21, no. 12, 1995, pp. 929 -944.

[11]   E. Dimitrov, R. Dumke, E. Foltin, and M. Wipprecht, "Conception and Experience of Metrics Based Software Reuse in Practice", International

Workshop on Software Measurement (IWSM), Lac Supérieur, Québec, 1999.

[12] N. E. Fenton and B. Kitchenham, "Validating Software Measures", Journal of Software Technology, vol. 1, no. 2, 1991, pp. 27-42.

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

[14] T. Fetcke, A. Abran, and R. Dumke, "A Generalized Representation for Selected Functional Size Measurement Methods", in Current Trends in Software Measurement - Proceedings of the 11th International Workshop on Software Measurement. Montréal: IWSM, Shaker-Verlag, Aachen; 2001, 3-8265-9681-1, 2001, pp. 1-25.

[15] H. Zuse, Software Complexity Measures and Methods, Walter de Gruyter, Germany, Berlin, 1991.

[16] H. Zuse, "Foundations of Validation, Prediction and Software Measures", Proceedings of the AOSW94 (Annual Oregon Software Metrics Workshop), Portland, 1994.

[17] H. Zuse, A Framework for Software Measurement, Walter de Gruyter, Germany, Berlin, 1997.

[18] B. Henderson-Sellers, Object-Oriented Metrics - Measures of Complexity, Prentice Hall, New Jersey, 1996.

[19] M. Hitz and B. Montazeri, "Chidamber and Kemerer's Metrics Suite: A Measurement Theory Perspective", IEEE Transactions on Software Engineering, vol. 22, no. 4, 1996, pp. 267 -271.

[20] IFPUG, "Function Point Counting Practices Manual", International Function Point Users Group (IFPUG), Version 4.1, 1998.

[21] ISO, "International Vocabulary of Basic and General Terms in Metrology", International Organization for Standardization - ISO, Geneva, 1993.

[22] ISO, ISO/IEC 14143-3:20033 "Software Engineering- Software Measurement - Functional Size Measurement- Part 3: Verification of Functional Size Measurement Methods", International Organization for Standardization ISO, Geneva, 2001.

[23] ISO, ISO/IEC19761:2003, Software Engineering -- COSMIC-FFP -- A Functional Size Measurement Method, International Organization for Standardization - ISO, Geneva, 2003.

[24] ISO ISO/IEC 9126-1 to 4: 2001-3, Information Technology-- Software Product Quality, International Organization for Standardization - ISO, Geneva, 2001 to 2003.

[25] 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.

[26] ISO, ISO/IEC 14143-5: 2003, Software Engineering - Functional Size Measurement- Part 5: Determination of Functional Domains for Use with

Functional Size Measurement, International Organization for Standardization - ISO, Geneva, 2003.

[27] J.M. Desharnais and P.M. Morris, "Post Measurement Validation Procedure for Function Point Counts", Forum on Software Engineering Standards Issues, Montréal (Canada), 1996.

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

[29] J.P. Jacquet and A. Abran, "Metric Validation Proposals: A Structured Analysis", in Software Measurement - Current Trends in Research and Practice, Proceedings of the 8th International Workshop on Software Measurement. Westdeutscher Verlag: IWSM 98, Deutscher Universität Verlag, Gabler Vieweg, 1999, pp. 43-59.

[31] M. Shepperd and D. Ince, Derivation and Validation of Software Metrics, Science Publications, Oxford, 1993.

[32] N.E. Schneidewind, "Methodology for Validating Software Metrics", IEEE Transactions on Software Engineering, vol. 18, no. 5, 1992, pp. 410-422.

[33] N.E. Schneidewind, "Body of Knowledge for Software Quality Measurement", IEEE Transactions on Software Engineering, vol. 35, no. 2, 2002, pp. 77-83.

[34] S.R. Chidamber and C.F. Kemerer, "A Metrics Suite for Object Oriented Design", IEEE Transactions on Software Engineering, vol. 20, no. 6, 1994, pp. 476-493.

[35] T.J. McCabe, "A Complexity Measure", IEEE Transactions on Software Engineering, vol. 2, no. 4, 1976, pp. 308-320.

[36] T.J. McCabe and C.W. Butler, "Design Complexity Measurement and Testing", Communications of the ACM, vol. 32, no. 12, 1989.