Validity Conditions Impact of a Software Measure

Miguel Lopez, Valérie Paulus

Abstract

Despite real progress in software engineering, the implementation of a successful measurement program in a software organization is still a serious challenge. Most of the problems are mainly due to the maturity level of the organization and to the credit given by the software engineering community for the software measure.

Software measurement, like other disciplines, must be based on the measurement theory; as such a scientific basis allows gaining a widespread acceptance among the community.

Validation of a software measure is an important task, but not an easy one. This task aims to prove that measures are actually measuring what they claim to do. One of the main problems is the relative validity of a software measure.

It is often observed that a software measure is strongly dependent on environmental factors. The goal of this paper is to highlight experimentally the impact of these factors, called validity conditions, on the validity measure.

Keywords : empirical validation, experimental validation,validity, measure.

Introduction

Nowadays only few measures are used in software organizations or departments. Maybe is this due to the fact that available measures are not scientifically validated. However, utilization of software measures in development process is also important to be able to improve quality and productivity [24]. In [11], Fenton affirms that :

> *(...) software measurement, like measurement in any other discipline, must adhere to the science of measurement if it is to gain widespread acceptance and validity.*

Moreover the theory of measurement [22] stress the importance of validate a measure. All these reasons (utilization, improvement, acceptance) underline the importance to work with valid measures. But what does 'valid measure' really means? First goal of this paper is to approach the concept of validity of a measure in more details than it is presented in general.

Validation of measure is a few documented subject. For example in [3] the validity of a measure is limited to the satisfaction of the condition of representation. In [2], a presentation of an experimental validation is made but nothing is proposed for the scientific viewpoint. [5] only talks about an empirical validation of measures. At the opposite, the theory of measurement [22] insists on the importance of validating a measure.

In previous work [16], [17], a measurement validation process has been defined. The aim of this process was to propose an easy way to scientifically validate a measure depending on the context of the measurement. The next step has been to wonder what could happen if the experimental environment ever changed? It seemed obvious that changes on the experimental environment would influence the result. But then a new question appeared. What is a change ? How can we experimentally highlight the influence of the environment on the validity of the measure ? Main goal of this paper is to present the impact that such a modification can have on the validity of a measure.

In section 2 we briefly introduce the Measure Validation Process defined in previous work. Then in section 3 we study the impact of changing the validity conditions on the validity of the measure with the example of response time as a measure of efficiency. Section 4 discuss the obtained results and we drawn some conclusions in section 6.

Previous Works : Overview of Measure Validation Process

Definition of Validation

Two kinds of measure validation exist in software measurement: the theoretical and the empirical validation.

(...) By theoretical validation we refer to the process of ensuring that metrics conform to the principles of measurement theory. By empirical validation we refer to the study of software in order to characterize, predict, control, manage or improve through qualitative or quantitative analysis [13].

The theoretical approach aims to clarify what software attribute we are measuring and how to measure those attributes [14], [12]. A measure must measure what it purports to measure. For example, in [11], the authors describes a set of mathematical properties that a measure of a object-oriented software attribute must satisfy and prove in particular that the Weighted Methods per Class (WMC) is a valid measure of the size.

In [5], Briand *et al.* affirm that *the application of the measurement theory in our field* [the software engineering] *is too rigid and even questionable.* Some prescriptions and proscriptions in software measurement (statistical techniques) are premature and prevent the progress of empirical software engineering.

Many authors have suggested empirical validation instead of theoretical validation. The goal of the empirical validation is to assess the usefulness and the relevance of any new measures [4], [7], [8]. In fact, the empirical validation consists on evaluating a measure in a number of case studies to build quality predictive models [22]. These models are able to predict quality attribute such as fault-proness and can be used for decision-making during a development project. Those validations are illustrated in Figure 1.

We strongly believe that both approaches can be combined. The Measure Validation Process we propose in previous work [17], is a way of validating software measure that integrate both empirical and theoretical validations. This validation is illustrated in grey in Figure 1.

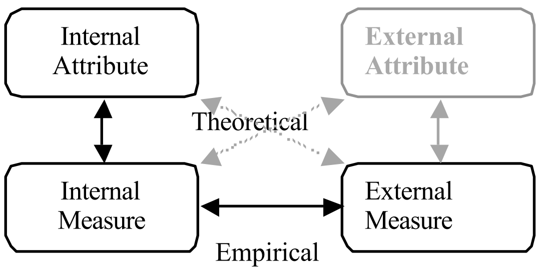

Figure 1 : Validation

Definition of Measure Validation

All the theoretical knowledge we have concerning empirical (observable) things make up a more or less complete picture of the empirical world. In other words, all the theoretical concepts together constitute another world, a world of theories and concepts. The idea of the theory as an image of the empirical world is called the scientific realism ????[KUH70]????, [9].

Researchers work simultaneously in these two worlds. According to this idea of a snapshot of the empirical world, the object of study exists in the empirical (tangible) world but theory belongs to the conceptual world of thinking. If the researcher accepts this view, he must provide for adequate connections between these two worlds. Unambiguous stated rules of correspondence are needed to link the theoretical model with the empirical object.

In software measurement, the object of study is the software product (defined as the set of computer programs, procedures, and possibly associated documentation and data [1]). The world of concepts is the mathematical world. The way of connecting these two worlds is provided by the Measurement Theory [22], [21].

Measurement theory specifies the conditions under which the empirical and numeric (mathematical) worlds can be combined. This theory translates empirical properties into mathematical properties [23].

In agreement with measurement theory and scientific realism, measure validity means that the observations (empirical world) match the concepts (theoretical or mathematical world). So, if a measure is valid, it measures what it intends to measure.

It is important to notice that a diversity of meanings are attributed to the term validity. No consensus has yet emerged, although it has prompted hard discussion in the software engineering community. There is a need for unambiguous definitions of the mathematical properties that characterize the major measurement concepts. Such a mathematical framework (measurement theory) could help to generate consensus among the software engineering community. The measure validity definition proposed in this paper is the following:

> *"A measure is valid if it satisfies a set of mathematical properties or axioms that model the attribute to be measured."*

The model (set of axioms) must capture the experts' knowledge concerning the software attribute by means of mathematical concepts defined in measurement theory [21].

Measure Validation Process

In previous work [17] we propose a methodology for defining valid measures or for validating existing ones. The aim of this section is to present briefly this methodology.

The methodology is compound of several steps:

- First we have to model the attribute we want to measure. This construction is mainly based on the intuition that practitioners have of the attribute and also on the state of the art for this subject. The empirical construction also consists in defining the representative sets on which the measure will refer. An empirical set is a behavior pattern of the attribute to model. It allows us to limit the scope of the measure to the field of the application to measure. More than simply define the representative sets, the practioner (named "expert" in previous work [17]) has to establish an order between each set in terms of the measure he wants to define. For example, if he wants to construct a new measure of performance, he has to indicates that the representative set A is more performant than the representative set B, and so on for all representative sets;

- Once we have this empirical structure, we could identify the numerical structure associated. This is compound of mathematical properties whose must be satisfied by the measure. Some properties came from the measurement theory. Others are based on intuition or state of the art. For example, it seems obvious that response time could not be negative;
- Define the measure in a way that the order we obtained by applying this one is strongly correlated with the order define in the empirical structure;
- Apply the measure on the Representative Sets and verify the correlation between both rankings (empirical and numerical). This step allows us to set the level of validity of the measure;
- Identify all factors that could influence the validity of the measure. These factors are called "validity conditions". And identify the real impact each of them has on the validity of the measure.

This methodology is illustrated in .

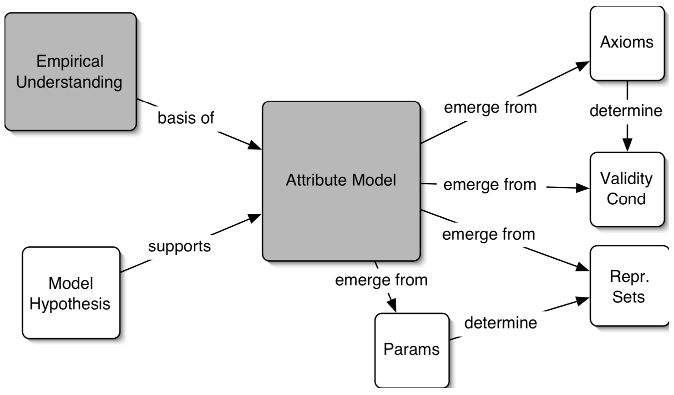

Figure 2 : Measure Validation Process

One of the main characteristics of the methodology is its openness. During the definition of empirical and mathematical construction, practioners can determine some working hypotheses that permit the evolution of the work. If during the process, one of this hypothese seems to be too strong or seems to be false, it is sufficient to identify the problem, to refine the hypothese or to soluce it and begin once again with repercuting changes along the process.

Main Features of Measure Validation Process

Contextual Validity

It seems obvious that a measure is only valid under specified conditions. In spite of that, this notion is not developed in software measure. The conditions of validity may be considered as all factors that influence the validity of the measure. In the Measure Validation Process, these are the precise conditions under which validation experiments have been carried out and which could invalidate the measure if a change of these conditions suddenly appears.

For example, in the case of a coupling measure[TBD:REF papier de darmstad], it is evident that the types of relation considered as coupling relation between two classes are major factors that influence the validity of the measure.

It is important to notice that a measure is valid if and only if it satisfies the set of mathematical properties that model the attribute. So, in order to determine the validity conditions and verify their exhaustivity, the question to answer is the following: which are the factors (conditions) that can have an impact on the satisfaction of each mathematical properties (axioms)?

Contextual validity allows to specify a scope that avoids an abusive generalization concerning the measure validity. This limitation prevents the use of the measure under conditions that can decrease its validity.

Open Process

In the definition of the validation process, some representative sets were defined by practitioners. These sets cover the most common attribute patterns that practitioners can meet. An assumption is made on the completeness of these sets. If a new set appears during the measurement phase, it must be integrated in the beginning knowledge and the process must be start again to validate the measure with this new knowledge. This empirical approach allows to start with a poor knowledge of the software attribute and to improve this knowledge by resolving problems when they appear. In that sense, the validation process presented here is an open procedure.

Efficiency Measure Validation

To prove the robustness of the methodology we try to validate response time as a valid measure of time behaviour – i.e a subcharacteristic of efficiency [TBD: ref ISO 9126]. So we play the role of practitioners and define an empirical structure. We propose a series of 8 representative sets which were PHP scripts having, for us, different time behavior. We give a ranking in terms of response time for all this scripts. Then we use a timer to effectively measure the response time of each scripts. Correlation between rankings we define in our empirical structure and this of the measure of response time shows us that response time is a valid measure of efficiency. The probability correlation between rankings is due to chance is of five percent. It means that the validity of the measure (correlation between rankings) is very strong. We also try to validate NLOC, executable statement and blanks in a source code as measure of efficiency. Intuitively it seems to be an erroneous measure, but we want to be certain that our process of validating measure was robust. And we could not obtain a sufficient correlation between rankings, so NLOC, executable statement and blanks are not valid measures of efficiency. More details about this experiment can be found in previous work [16]

Validity Conditions Impact

Presentation of Validity Conditions and Mathematical Properties

In [16],[17] we present the validation of response time as a measure of time behavior (efficiency). The identified numerical structure contains 5 mathematical properties the measure has to satiffy. These are the following:

- Homomorphism

- Strong completeness
- Transitivity
- Non negativity
- Null value

In this context, we also identify a series of validity conditions (factors that can influence the satisfaction by the measure to one of these mathematical properties). Table 1 shows the validity conditions of the response time of a web application. The validity conditions are grouped in four different categories: the platform, the software, the network and the processor activity. Practioners assume that each element of the four categories has an impact on the validity. They determine the validity conditions following their experience and their knowledge of the domain. Practioners have also the intuition that each validity condition has a real impact on the measure validity. In this case, real means that an experiment can highlight the impact of the validity condition on the measure validity.

This paper is focused on the platform, the network, the processor activity and the measurement tool. It is assumed that these conditions are the most influent on the validity and that the Software has a low impact on the measure validity except for the measurement tool.

Conditions Validity Description	
Plateform	Hard Disk, OS, Processor, RAM
Software	HTTP Server and Client,DBMS, Language, Measurement Tool
Network	Ethernet Card, Bandwith
Processor Activity	CPU Load

Table 1 : Validity Conditions

Three mathematical properties seems to be critical for the validity of response time in regard with the previous validity conditions :

- The homomorphism
- The null value
- The non negativity

However, the mathematical property of weak order (strong completeness and transitivity) only concerns the representative sets and the set of reals. The validity condition does not impact the respect of the axiom of weak order of the set of reals. If conditions are changed, the set of reals [TBD: pourquoi ce signe ?]< (that is the set of the numerical order) will preserve the properties of weak order. The same can be affirmed concerning the set of paradigms.

Estimated Impact

Homomorphism

As expressed before, practitioners ranked the representative sets regarding the time behavior. The ranking is a list of paradigms from the most rapid program to the slowest one. This order is called the empirical order. The practitioners find out the empirical order by consensus. Their judgment must reflect their understanding of the attribute to measure, i.e. the time behavior. This judgment can take into account the validity conditions, but is based on the intuition of the experts.

Another order is considered in this procedure, the mathematical order given by the measure. The measure defines a mathematical or numerical order. It is expected that a modification of one of the validity conditions would have an impact on the measure and change the numerical order. If the correlation between the empirical order and the new numerical order is weak or null, the measure is invalid.

Null value

The null value axiom is used to verify whether the measurement tool is correctly calibrated. It is assumed that the response time of an empty script (without any statement) equals zero second. This assumption is false. In the real world, the processing of an empty script will take some time, even if it is not greater than 10^{-5} second. From this point of view, the calibration of the measurement tool is made of two steps:

• The determination of the response time of the empty script.

• The correction of the other response times of the scripts regarding the empty script response time.

This response time of the empty script is considered as the *experimental zero*. The assumption that is expressed here is the following: the modification of one single validity condition could violate the null value axiom. This would mean that the measurement tool is defective under the new validity conditions. Consequently, the measure is invalid because of the defectiveness of the measurement tool. But, maybe another measurement tool would work correctly and give valid measure under these new conditions. It is important to notice that this axiom is not precisely proved by the experiment. In fact, the satisfaction of this axiom is forced. In other words, in order to satisfy the null value axiom, the measured value for the empty script is defined as the experimental zero, i.e. the first step of the calibration.

Non negativity

The non negativity axiom asserts that the response time is greater than zero. It is not possible that a response time would get a negative value. The measurement of a negative value could mean two different things:

• before calibrating: the measurement tool does not work correctly under the new conditions.

• after calibrating: the paradigms for which the response time is negative are not differentiable from the empty script

So, it is important to notice that a measure can not be separated from its measurement tool. A measure is also invalid if its measurement tool works defectively. A response time measurement tool is valid if the null value axiom *or* the non negativity axiom are satisfied. The validity of the measurement tool will be the purpose of future works.

Tested Hypothesis

The goal of this section is to describe the different hyptoheses we try to verify by experimentation. The global aim remains the impact of a change of the validity conditions on the validity of a measure.

Table 2 shows the high level hypotheses tested during the experiment phase. Table 3 shows lower level assumptions which where derived from those of Table 2 in order to experimentally test them. The notation of the low level hypotheses refer to the corresponding high level ones, i.e. H_{11} refers to H_1.

Hypothesis	Description
H_1	The platform does not influence the validity of the measure response time
H_2	The low traffic network does not influence the validity of the measure response time
H_3	The high traffic network influences the validity of the measure response time
H_4	The processor activity influences the validity of the measure response time

Table 2 : High Level Hypotheses

Hypothesis	Description
H_{11}	Response Time is a valid measure of a Web application in PHP under Windows 2000
H_{21}	Response Time is a valid measure of a Web Application in PHP whose server is connected to a local area network
H_{41}	Response time is an invalid measure of a Web Application in PHP in multiple connected clients context

Table 3 : Low Level Hypotheses

The hypothesis H_{11} express the fact that we modify the operating systems on wich the web application in PHP run to change the validity condition. H_{21} should show us the difference that can exist in a network or stand alone environment. And third, hypothesis H_{41} show the influence of a increasing number of clients.

Experiment and Results

Global results and execution of the experimentation can be found in [16]. The goal of this section is to present sufficient informations on the results of experimentation for the reader understanding.

Platform

Object of the study: the platform as a validity condition of the response time that is a measure of the time behavior in terms of efficiency.

Purpose: the purpose is to verify whether the platform modification (from Mac OS X to Windows 2000) can influence the validity of the response time.

Quality focus: the quality focus is the influence of platform Windows 2000 on the validity of the response time.

Perspective: the perspective is from the researcher's point of view.

Context: the experiment is run in one standalone computer (Intel PIV 1.5 Ghz, 512 RAM) using a software tool for measuring the response time of 8 programs.

Network

Object of the study: the network as a validity condition of the response time that is a measure of the time behavior in terms of efficiency.

Purpose: the purpose is to verify whether the network modification can influence the validity of the response time.

Quality focus: the quality focus is the influence of network on the validity of the response time.

Perspective: the perspective is from the researcher's point of view.

Context: the experiment is run in a server machine (Intel PIV 1.5 Ghz, 512 RAM) using a software tool for measuring the response time of 8 programs and a client machine (PPC G3 800 Mhz 256 RAM) that launches the text-based browser Lynx.

Processor Activity

Quality focus: the quality focus is the influence of processor activity on the validity of the response time.

Perspective: the perspective is from the researcher's point of view.

Context: the experiment is run in a server machine (Intel PIV 1.5 Ghz, 512 RAM) using a software tool for measuring the response time of 8 programs and a Apache Jmeter v1.8.1 that simulates the simultaneous connections. Jmeter is installed on a separated machine (PPC G3 800 Mhz, 256 RAM).

Table 4 shows the satisfaction of the mathematical properties regarding the validity conditions. Mathematical properties are labeled as follow :

- MP_1 :the preservation of relation oder (homomorphism)
- MP_2 :the non negativity property
- MP_3 :the null value property

The symbol √ means that the mathematical property is satisfied and the symbol × means that the mathematical property is unsatisfied. Correlation is given by the first value in the parenthesis and the level of confidence is the second one.

Conditions	**MP_1**	**MP_2**	**MP_3**
Platform	√(0.88095 ; 10%)	√	√
Network	√(0.9286 ; 5%)	×	√
1 Client	√(0.8571 ; 10%)	×	√
5 Clients	√(0.8809 ; 10%)	√	√
10 Clients	√(0.8809 ; 10%)	×	√
15 Clients	√(0.9047 ; 5%)	√	√

Table 4 : Mathematical Properties

Discussion

The non negativity axiom is not satisfied in three cases (Network, 1 client and 10 clients). None of the measured values were negative. The negative values appeared after calibrating the response time average, regarding the response time average of the empty script. Notice that the problem always occurs with the same paradigm, which is A3 (a PHP script with 2 instructions that declares a session variable and assigns the value of an environment variable (HTTP_ACCEPT_LANGAGE). The script A3 has a response time average less than the empty script for three cases (see tables 10 and 12).

Several explanatory assumptions can clarify this situation.

1. Some errors occurred during the measurement: errors of measurement, errors of calculations,...
2. The ranking of A3 in the empirical order (given by experts) is not correct, i.e. the A3 response time is less than the experimental zero (due to the measurement tool sensibility, which is defined as the minimum delta detectable by the measurement tool.)
3. The proposed calibration method is not correctly defined.
4. The Response Time is not a valid measure of the time behavior of a Web application in PHP.

Rejected explanatory hypothesis The first hypothesis, which states that errors have been made during the measurement, is rejected due to the repetition of the measurements (5 times with 1000 measurements for each script). The last explanatory hypothesis, which states that the response time is not a valid measure, is not taken into account because it is accepted and obvious that the response time is a valid measure.

Accepted explanatory hypothesis The second and the third assumptions are considered as serious solutions to the problem. So, the future works will verify the second assumption which states that the third representative set (A3) is erroneous, and the third assumption which aims to refine the calibration method. In this context, the wrong ranking of A3 invalidates the working hypothesis which states that the empirical order is correct.

Conclusion

Table 5 shows the measured impacts for each group of validity conditions.

Notice that the expected results only match with the experimental result of the Platform. The low traffic network and the processor activity appeared to have a relative influence on the validity.

How to interpret these more or less influence of the Network and the Processor Activity on the validity ? At the beginning of this work, the validity and validity conditions influences were defined as boolean concepts.

On the one hand, it was accepted that the a measure can be valid or invalid. But, in this work, the validity seems to be a more fuzzy concept. In other words, a measure has a certain level of validity which depends on the validity conditions. This is why and how the validity of the measure in a network context can not be rejected.

On the other hand, the influence of the validity conditions on the measure validity was modelled as a boolean impact (influent/not influent). The fact that under some conditions the axiom 2 (see table 14) is not satisfied and that under some other conditions the level of confidence of the coefficient correlation equals 10% are not sufficient rationale for invalidating the response time. It is accepted that the response time is a valid measure of the efficiency of a Web application. If the previous assertion is true, the only way for clarifying the results is to define the validity and the influence of validity conditions as fuzzy concepts. These new definitions will be the purpose of a future works.

Validity Conditions	**Measured Impact**

Plateform	Not influent
Network with low traffic	+/- influent
Network with high traffic	-
Processor Activity	+/- influent

Table 5 : Results Experiments

The validity of the response time is hardly tied to the context, which is defined as the conditions in which the validity of a measure is tested. Further works will refine the understanding of the influence of the validity conditions on the measure validity, i.e. the response time.

Acknowledgement

That research project is supported by the European Union (ERDF) and the Walloon Region (DGTRE) under the terms defined in Convention n°EP1A12030000072-130008.

References

1. Jmeter documentation, 2001.
2. ABRAN, A. Metrics validation proposals: A structured analysis. *8th International Workshop on Software Measurement* (1998).
3. ALEXANDRE, S. Software metrics. an overview. Technical report, Institute of Computer Sciences. FUNDP, Namur, Belgium, 2002.
4. BAKKEN, S. S. PHP documentation, 2003.
5. BRIAND, L., DALY, J., PORTER, V., AND WST, J. A comprehensive empirical validation of product measures for object-oriented systems, 1998.
6. BRIAND, L., EMAM, K. E., AND MORASCA, S. On the application of measurement theory in software engineering. Tech. Rep. ISERN-95-04, Centre de Recherche Informatique de Montr´eal, 1801 McGill College Ave., Montreal PQ, H3A 2H4, Canada, 1995.
7. BRIAND, L., EMAM, K. E., AND MORASCA, S. Theoretical and empirical validation of software product measures. Tech. Rep. ISERN-95-03, Centre de Recherche Informatique de Montr´eal, 1801 McGill College Ave., Montreal PQ, H3A 2H4, Canada, 1995.
8. BRIAND, L. C., MORASCA, S., AND BASILI, V. R. Property-based software engineering measurement. *Software Engineering 22*, 1 (1996), 68–86.
9. CHALMER, A. Science and Its Fabrication. Open University Press and University of Minesota Press, 1990
10. EMAM, K. E. A methodology for validating software product metrics.
11. FENTON, N., AND PFLEEGER, S. L. *Software Metrics - A Rigorous and Practical Approach*, 2 ed. International Thomson Computer Press, London, 1996.
12. FENTON, N. E. Software measurement: A necessary scientific basis. In *PDCS 2: Open Conference* (Newcastle-upon-Tyne, 1994), Dept of Computing Science, University of Newcastle, NE1 7RU, UK, pp. 429–446.
13. ISO. Software engineering-product quality part 1 quality model, 1999.
14. ISO. Software engineering-product quality part 2 external metrics, 1999.
15. KITCHENHAM, B., PFLEEGER, S., AND FENTON, N. Toward a framework for software measurement validation. *IEEE Transactions on Software Engineering 21*, 12 (Dec. 1995), 929–944.
16. LOPEZ, M., PAULUS, V., AND HABRA, N. Towards a validation process for the measure of the efficiency: integrating axiomatic and empirical approaches. Tech. Rep. LQL-2003-TR-01, University of Namur, Avenue Grangagnage, 21 5000 Namur, Belgium, 2003.
17. LOPEZ, M., PAULUS, V., AND HABRA, N. Integrated Validation Process of Software Measure. In *Proceedings of the International Workshop on Software Measurement (IWSM 2003*) (2003)
18. MELTON, A. C., GUSTAFSON, D. M., BIEMAN, J. M., AND BAKER, A. L. A mathematical perspective for software measures research. *Software Engineering Journal 5*, 5 (Sept. 1990), 246–254.
19. MORASCA, S., BRIAND, L. C., BASILI, V. R., WEYUKER, E. J., AND ZELKOWITZ, M. V. Comments on “Toward a framework for software measurement validation”. *IEEE Transactions on Software Engineering23*, 3 (Mar. 1997), 187–188.
20. POELS, G. Towards a size measurement framework for object-oriented specifications.

21. PREFECT, E., DOAN, L., GOLD, S., WICKI, T., AND WILCKE, W. Performance limiting factors in http (web) server operationss. In *Proceedings of the COMPCON 41st IEEE International Computer Conference* (1996), pp. 6390–96.
22. ROBERTS, F. S. *Measurement Theory with Applications to Decisionmaking, Utility, and the Social Sciences*, 1 ed. Addison Wesley Publishing Company, 1979.
23. SCHNEIDEWIND, N. F. Methodology for validating software metrics. *IEEE Transactions on Software Engineering 18*, 5 (May 1992), 410–422.
24. SEDAYAO, J. World Wide Web network traffic patterns. In *Proceedings of the COMPCON '95 conference* (1995).
25. WOHLIN, C. *Experimentation in software engineering : An introduction*, 1 ed. Kluwer Academic Publisher, 2000.
26. ZAHRAN, S. *Software Process Improvement – Practical Guidelines For Business Success*, 1 ed. Addison Wesley Publishing Company, 1998.
27. ZUSE, H. Validation of measures and prediction models. In *Proceedings of the International Workshop Software Measurement (IWSM'99)* (1999).