Towards a validation process for the measure of the efficiency: integrating axiomatic and empirical approaches.

Miguel Lopez[1], Valérie Paulus[1], Naji Habra[2]

[1]CETIC asbl, Rue Clement Ader, 8
6041 Gosselies Belgium
{malm,vp}@cetic.be
http://www.cetic.be

[2] University of Namur, Rue Grandgagnage, 21
5000 Namur Belgium
nha@info.fundp.ac.be
http://www.info.fundp.ac.be/~software-quality

Abstract

The validity of the measures used in software engineering is a critical matter about which no consensus could be reached at this point, though hard discussion. There is a need for unambiguous definitions of the mathematical properties that characterize the major measurement concepts. Such a mathematical framework could help generate a consensus among the software engineering community.

The goal of this paper is to provide a formal validation process for software measure. It presents a global measurement framework that integrates theoretical and empirical validation processes based on the measurement theory. The concept underlying the framework is to formalize some properties of the measure to analyze and then to verify the conformity of these properties to the measure thanks to a formal experiment.

This validation process determines a contextual validity (scope) defined by the set of factors or validity conditions that impact the validity of the measure. The paper develops a case study that validates, under specified conditions, the response time as a measure of the efficiency as defined by the ISO/IEC 9126 standard.

Keywords: *Axiomatic validation approach, empirical validation, formal validation, measurement theory, web applications, web metrics, contextual validity, efficiency, response time, ISO/IEC 9126 standard.*

1. Introduction

The aim of this paper is to give an outline of how to validate a measure based on the principles of measurement theory. A measure validation is defined as "the process which controls that a measure represents correctly the attribute it has to measure" [FENT96].

The validity of the measures used in software engineering is a critical matter about which no consensus could be reached at this point, though hard discussion. There is a need for unambiguous definitions of the mathematical properties that characterize the major measurement concepts. Such a mathematical framework could help reaching a consensus among the software engineering community.

The goal of this paper is to provide a formal validation process for software measure. It presents a global measurement framework that integrates theoretical and empirical validation processes based on the measurement theory. The concept underlying the framework is to formalize some properties of the measure to analyze and then to verify the conformity of these properties to the measure thanks to a formal experiment.

This validation process determines a contextual validity (scope) defined by the set of factors or validity conditions that impact the validity of the measure. The paper develops a case study which validates, under specified conditions, the response time as a measure of the efficiency as defined by the ISO/IEC 9126 standard

This paper is organized as follows. First, in Section 2 some definitions of measure validity are explained. At the end of the Section, a new definition for the measure validity is proposed. Next, in Section 3, a validation procedure based on the validity definition proposed in the Section 2 is explained. This validation procedure is a twofold approach: first the specifications of the empirical and mathematical system are given; and then, the validation of these two systems is done theoretically or experimentally. This is followed in Section 4 by the experimental validation of the preservation of the empirical order in the mathematical system. Lastly, a conclusion is made with some comments about the future research directions.

2. The measure validity

2.1. The lack of consensus

Bieman's definition.

In [BIEM92], a definition of a valid measure is given: "a software measure is only valid if it can be shown to be an accurate predictor of some software attribute". This

definition highlights the necessity to know what the measure really measures before its validation.

Representational condition

Another approach for validating a measure is based on the representational condition. *A measure is valid if it satisfies the representation condition: if it captures in the mathematical world the behavior we perceive in the empirical world* [FENT96]. If Mr A is taller than Mr B and if μ is a measure of the human size, then μ(A) should be greater than μ(B). The order perceived in the empirical world (Mr A is taller than Mr B) must be kept in the mathematical world (μ(A) is greater than μ(B)). This is a necessary but insufficient condition for validating the measure. The measure must satisfy other kinds of properties. For example, the measure of the size must be positive. So, a validation based only on the representational condition would not validate correctly the measures. This kind of validation does not take into account the understanding (the model) of the attribute to measure. In the above example, the measure of the human size must be positive. The model of the human size assumes that this measure is positive. In this sense, the representational condition is not enough.

IEEE Definition

In [IEEE93], a valid measure is defined as a measure *whose values have been statistically associated with corresponding quality factor values*. This definition gives a necessary condition which is insufficient. In [ZUSE99], H. Zuse affirms that *Using only statistics without knowing the models* (understanding the attribute) *behind valid measures and prediction models does not lead to solid results.* The software engineer needs attribute models based on environment hypothesis to validate his/her measures. The understanding of the software attributes is not accepted by the scientific community. For example, the behavior of a program with respect to the operation of concatenation is prone to controversy. Is the program P made of the concatenation of the programs P_1 and P_2 is more efficient than the separated programs P_1 and P_2 ? The answer is not obvious. There is a lack of consensus concerning the software attributes models.

Necessity of a consensus

This set of three definitions is not exhaustive but proves the diversity of the significances of the term validity. It is important that the scientific community together with the industry agree about the validation meaning. The software engineers will use the measures if and only if they can work with a serious and solid definition of the validity. A serious and solid validation process can be based on the measurement theory. The measurement theory is a mathematical framework that would facilitate the consensus among the experts (scientists and the industry).

2.2. Hypothesis on the environment

In [HEND96], Henderson-Sellers affirms that *(...) many metrics are validated against only one data set. This does not, in itself, render the validation process invalid but cannot be used to justify anything other than a very restricted and careful use of the metric.* The validation process is done under specific conditions that can impact the measure validity. These conditions can have such an impact on the validity that the modifications of one of them can invalidate the measure. It is important to describe precisely the conditions of the validation and to enjoin the user of the measure to verify the conditions of his/her environment.

The conditions of validity are in fact the hypothesis of the software attribute model. These assumptions capture the whole understanding of the attribute. In the example of the software efficiency: the current understanding assumes that an empty program (without any statement) must have a null response time. This assertion is an hypothesis of the software efficiency model. It is necessary to establish a complete model regarding the state of the art.

2.3. The experts' knowledge

Experts build the model of the software attribute to be measured. This model captures their understanding of the attribute for which a consensus is reached. This model includes the validity conditions, their impacts on the validity and the properties of the measure (empirical order preservation, positivity…). The model is an evolutionary one, it must often be revised.

2.4. Definition proposal

A measure is valid if it satisfies a set of mathematical properties or axioms that model the attribute to be measured. Therefore, a group of experts must establish a set of axioms by consensus. The representational condition is a mandatory property for all measures. In fact, this is not true for all properties, i.e. the non-negativity above. The model must capture the experts' knowledge concerning the software attribute by means of mathematical concepts defined in the measurement theory [ROB79]. This theory provides a rigorous framework to the software engineer and would facilitate the establishment of a consensus among the experts.

3. Proposal for a Validation Procedure

3.1. Goals of measurement

The goal of the measure to be validated as an example of this framework is to allow the comparison of pairs of programs in terms of software efficiency. The valid measure must allow affirming that a program A is faster,

thus more efficient, than a program B. To do this, we refer to the efficiency definition given in the ISO/IEC 9126 standard [ISO9126]: *"The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions."*

3.2. Specification of the empirical relational system

*Characterization of the measured attribute***.**

In this study, we are particularly interested in time behavior of the software product. In the quality model of the ISO/IEC 9126 standard, time behavior is a sub-characteristic of efficiency and is defined as follow: *"The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions."*

The measurement theory specifies conditions under which combination between empirical and numeric world can be made. This theory translates empirical properties into mathematical properties. [ZUSE97]

According to the measurement theory, in our case, we consider:

- The set ***A***: which is composed of the set of all n-tiers Internet architecture programs in PHP.
- A binary relations R in ***A***: a relation between programs is needed to express the comparison.

The set of pairs R includes all pairs of programs related by 'have a more adequate behavior than'. In the case of set ***A***, we suppose that this relation is equal to 'is faster than'.

With the set ***A*** and the relation R, we have an empirical relational system, called *U*=(***A***,R). We claim that this system is empirical because it regroups entities of real world.

Now, in order to refine our definitions, let us propose:

- ***A*** : the set of n-tiers Internet architecture programs in PHP
- R : the relation 'is faster than' symbolized by "**>**"

*The set **A***

The set ***A*** could not be described exhaustively and its complete description seems not to be useful. We define ***A*** with paradigms, i.e. ***A*** includes all web based applications with real case, for example:

- An empty PHP script: with no line of code
- A PHP script which executes a set of instructions without communication towards a third party application
- A PHP script which communicates towards a third party application

Practically, we can not deal with all the web-based applications of a domain. The idea is to choose a representative sample of the theoretic set ***A*** in function of the sub-characteristics we want to measure. Let us write the representative set: set A. In other words, as the validation object is the measure of the time behavior, it is obvious that the chosen set A will include scripts which suitably represent the efficiency problem.

The set A is a set of real world objects but these objects are not practical ones. They represent different typical situations met in practice. For this reason, experts must specify the set A. It is here assumed that the set of paradigms is representative of the problem of the n-tiers Internet architecture. This is to be considered as a working hypothesis for this paper whose goal is to present a validation process and to validate internally a measure of the efficiency.

The relation

By experience, we know that if a PHP script A_1 contents three instructions and if a PHP script A_2 contents A_1 and a bloc of instruction executing a SQL request, then we could say that:

$$A_1 > A_2 \quad (1)$$

So now we have:

- A : a set of n-tiers Internet architecture application paradigms for a given domain
- R : the relation "is faster than", "**>**" on A
- *U* : (A, **>**) : a relational empirical system

Specification of system U.

The set A has a finite number of elements. To have the possibility to compare pairs of programs, the set A must be an ordered set, in other words a weak order. To have a weak order the following axioms have to be satisfied:

- Strongly complete: $\forall a,b \in A : a > b \vee b > a$ (2)
- Transitive: $\forall a, b, c \in A : a>b \wedge b>c \Rightarrow a>c$ (3)

In other words, the axiom (2) means: for any pair of programs (a,b) from set A, either a is faster than b or b is faster than a. The axiom (3) can be read: if a is faster than b and b is faster than c then a is faster than c. The satisfaction of these two axioms allows comparison between elements of the set A.

Let us consider A to be the following finite set of PHP web based paradigms described in an extensive way:

$A = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\}$

A_1 = PHP 4.2.1 script without any instruction and a white blank.

A_2 = PHP 4.2.1 script with 4 instructions: a test *if* on two integers.

A_3 = PHP 4.2.1 script with 2 instructions that declares a session variable and assigns the value of the environment variable HTTP_USER_LANGUAGE.

$A_4 = A_2 + A_3$: PHP 4.2.1 script containing A_2 and A_3

A_5: write one line (a real number) in a text file

$A_6 = A_3 + A_5$: writes in a text file and declare a session variable

$A_7 = A_3 + A_5$: the same script as A_6 but with a loop that writes 1000 times a line in text file.

A_8: the same script as A_7 but instead of 1000 times, it writes 5000 times the same line.

The elements of set A are chosen in order to avoid the verification procedure to be affected by the problem of the measurement tool sensibility.

The relation *R* can be written as follows:

$$A_1 > A_2 > A_3 > A_4 > A_5 > A_6 > A_7 > A_8 \qquad (4)$$

This definition of the system *U* must be given by experts, who will establish the sequence expressed in (4) by consensus. We prefer another notation for the set *R* :

$$R = \{(A_1, A_2), (A_2, A_3), (A_3, A_4), (A_1, A_3), (A_1, A_4), (A_2, A_4), \ldots, (A_6, A_7), (A_6, A_8), (A_7, A_8)\} \quad (4a)$$

The expert who made this classification is confronted to scripts easy to classified from a response time point of view. It is assumed that the distinction between two scripts, which differ from each other with a single instruction, is more a problem of measure instrument sensibility than a problem of judgment by the experts. New elements can be added in A, this set is not exhaustive.

3.3. The group of experts.

The aim of this section is to introduce the problem of the expert choice.

Role of the experts.

The determination of the empirical system, i.e. the specification of sets A and R, are under the competence of experts in the domain of PHP web based applications (in this example). The job of the experts is to express, by means of mathematical properties, the comprehension they have of time behaviour. Clearly, these mathematical properties are the above quoted axioms. It is assumed that these axioms represent the actual state of our knowledge in matters of time behaviour in PHP web based applications.

Choice of the experts.

The choice of the experts introduces the problem of mutual acknowledgement. We can choose a group of k experts in our quality software laboratory. The problem that appears is to know if the expertise of our experts will be recognized by other experts and by users of measure. Because we claim that the list of axioms is exhaustive (in the current state of our knowledge), the choice of experts will be critical for the validity of measures. Question is to know if the measures validation can be done in private or in the framework of an industrial and scientific community?

Take a software development company C_1 who wants to validate some measure in its context (i.e. in their own validity conditions). C_1 must select some experts to determine the axioms that have to be validated. Take another company C_2 that wants to use the measures validated by the first one. The measure's context of use is the same in C_2 than in C_1. If the management of C_2 does not know the group of experts selected by C_1, how can C_2 trust the results of the validation made in C_1 ?

In general, how can we trust the completeness and the relevance of axioms determined by experts we don't know. The following use case describes a validation realised privately. If the aim is to reach a consensus by industrial and scientific community, it's obvious that the private validation is purely anecdotic. We strongly believe that to obtain a global consensus, the specification of axioms must be led at an international level by group of experts. For us, the ISO organisation is an adequate structure for such a work.

3.4. Specification of the numeric relational system

*Mathematical assignation rules***.**

The goal of this section is to define the measure of time behavior. The ISO/IEC 9126 standard gives us as time behavior measure: the *response time*.

Based on the measurement theory, we could consider:

- B : the set of the reals, Re
- R : a binary relation in B

So we have *B* (B,R) a numeric relational system. In this one we could replace B by Re and R by relations "is smaller than" or "<".

In the measurement theory, a measure is defined as a function from *A* toward *B* that preserves the relations from system *U* into system *B*. This function is called homomorphism. It's a mapping between the empirical relational system and the numeric relational system. If μ

is a measure, the homomorphism can be expressed as follows:

$\mu : A \rightarrow B : \forall a, b \in A : a > b \Leftrightarrow \mu(a) < \mu(b)$ (5)

If we consider that μ, in the expression (5), represents response time, we could state that this expression is an axiom. The axiom (5) says that if, empirically, the experts note that the program a is faster than the program b, then the measure of response time of a is smaller than the measure of response time of b a and vice versa. If the measure of response time of a is smaller than the measure of response time of b, then a is faster than b. This axiom ensures that the relation in the empirical world is preserved in the numerical world.

Specification of system B.

The system *B* is a numerical relational system. The measure has to satisfy axioms (2) to (5) because μ must be a homomorphism.

$\mu:A \rightarrow B : \forall a,b \in A : a>b \Leftrightarrow \mu(a)<\mu(b)$ (5)
Strongly complete: $:\forall a,b \in A : a > b \mathbf{v} b>a$ (2)
Transitive: $\forall a,b,c \in A : a>b \wedge b>c \Rightarrow (a>c)$(3)

Table 1: The axioms

The axioms (2) and (3), by homomorphism, can be written as follows :

- Strongly complete:$\forall \mu(a),\mu(b) \in B : \mu(a) > \mu(b)$ v $\mu(b)>\mu(a)$ (2a)
- Transitive: $:\forall \mu(a),\mu(b),\mu(c) \in B: \mu(a)>\mu(b) \wedge \mu(b)>\mu(c) \Rightarrow (\mu(a)>\mu(c))($ (3a)

$\mu:A \rightarrow B : \forall a,b \in A : a>b \Leftrightarrow \mu(a)<\mu(b)$ (5)
Strongly complete:$\forall \mu(a),\mu(b) \in B : \mu(a) > \mu(b)$ v $\mu(b)>\mu(a)$ (2a)
Transitive: $\forall \mu(a),\mu(b),\mu(c) \in B : \mu(a)>\mu(b) \wedge \mu(b)>\mu(c)) \Rightarrow (\mu(a)>\mu(c))$ (3a)

Table 2: The revised axioms

3.5. Specifications of the measure instrument

The null value.

The null value: $\forall a \in A : a = \emptyset \Leftrightarrow \mu(a) = 0$ (6)

The axiom (6) is not necessary to compare pairs of programs. To do this, only a measure of time behavior which allows us to compare programs (i.e. to confirm that a program a is faster than a program b) is needed. But to obtain a valid measure, we should take this axiom into consideration, so μ has to respect the axiom (6)

Non-negativity.

Non negativity: $\forall a \in A : \mu(a) > 0$ (7)

It's the same for the axiom (7). The respect of axiom (7) by the mapping μ is a way to prove that if we obtain a negative measure, then there is a mistake in the measurement process (for example the instrument of measure is defect).

3.6. Validation of system *U*

Strongly complete.

Strongly complete: $\forall a,b \in A : a> b \mathbf{v} b>a$

Definition of R given by (4a) respects the axiom (2a). All possible pairs are represented in the set R and defined based on the relation ">". Thus the set A based on the relation ">" is, by definition, strongly complete.

Transitive.

Transitive: $\forall a,b,c \in A : a>b \wedge b>c \Rightarrow a>c$ (3)

Let the set A as defined in (4) and the set of relations R defined in (4a). The axiom (3) affirms that for all scripts a, b and c from A if a is faster than b and b faster than c then a is faster than c.

Proof: Let take each pair of A defined following the relation">" and verify that transitivity is respected.

$$A_1>A_2, A_2>A_3 \Rightarrow A_1>A_3 \quad (8)$$

Based on the relation R defined in (4a), if A_1 is faster than A_2 and A_2 is faster than A_3 then A_1 is faster than A_3. The same comparison can be done for all elements of the set A. The results of each comparison confirm the axiom of transitivity.

The set A based on the relation ">" is thus transitive. Transitivity is respected by all elements of set A.

3.7. Validation of the measurement instrument

Null value.

If the script a is empty, then the measure μ(a) is null. If, during the measurement phase, a case appears where the script a is empty but the measure μ(a) is not null then it could be assumed either that the measure μ is not valid or that an error occurs during the measurement phase as, for example, a wrongly calibrated measure instrument. This axiom is not subject to verification but it ensures the validity of the measure during the measurement phase.

Non-negativity.

The measure μ is positive. In the case of a negative measure, we affirm either that the measurement phase is erroneous or that the measure μ is not valid. Once again, non-negativity is an axiom which will not be verified but which ensures to the user of measure a certain validity of the collect of measures sample.

3.8. Validation of system B

Strongly complete.

The system *B* satisfies the axiom (5a) because *B* is the set of reals that is strongly complete. We could affirm that if μ(a) and μ(b) are real then either a is strictly smaller than b or b is strictly smaller than a.

Transitivity.

If μ(a), μ(b) and μ(c) are real and μ(a) is smaller than μ(b) and μ(b) is smaller than μ(c) then μ(a) is smaller than μ(c). The transitivity is well satisfied by the set of reals.

Relations.

Axiom (2) requires the preservation of the relations between the measures and the attribute of observed object. If the script a is faster than the script b, then the measure has to reflect this relation, i.e. μ(a) should be smaller than μ(b). Verification of axiom (2) is based on a formal experiment. Following a collect of results, a correlation test of Spearman must be done.

The rank correlation coefficient (Spearman coefficient) measures the relation between rank of observations of two characters X and Y. This coefficient allows detecting the existence of monotone relations (increasing or decreasing).

To know if a relation is significant, a test of hypothesis must be done as follows:

- Null hypothesis, H_0: there is no relation between the attributes X and Y.
- Fix a level of significance for rejecting the null hypothesis (e.g. $\alpha = 5\%$).
- Calculate the value of the Spearman coefficient r(X,Y).
- Calculate the theoretical value of the Spearman coefficient $r(\alpha,n)$ with the statistical tables, n is the degrees of freedom.
- Test H_0 true if $r(\alpha,n) > |r(X,Y)|$.
- Accept or reject H_0.

The experiment and the results are described in detail in the section 4. The structure of the experiment is based on [WOHL00].

3.9. Conditions of validity

Proposed definition.

The conditions of validity can be as all the factors that influence the validity of the measure. These are the exact conditions under which the axioms validation experiments have been realised and which could lead to an invalidation of the measure. An invalid measure is a measure, which does not satisfy one of the axioms of table 2. For example, in the case of response time, it's obvious that processor speed and RAM quantity are major factors which influence the validity of the measure.

List of validity conditions.

The definition of response time specified in the ISO/IEC 9126 standard contents two parts: execution time and time of command entry. In this example, the time of command entry can be disregarded, i.e. considered as null, because it's a FOR loop that triggers the execution of the script. The formula of response time is then reduced to the execution time.

It is assumed that each factor of the table 3 has an influence on the validity of the measure of the response time.

1	Processor: frequency, type (Intel, PPC, …)
2	RAM: frequency, type, quantity
3	Hard disk: type, capacity (if disk access)
4	Operating system (version)
5	Network connectivity: web, LAN, …
6	Programming language
7	Apache version
8	Browser: version, type
9	Developer maturity
10	Active process (CPU load)
11	Number of connected clients
12	Number of requests

Table 3 : Conditions of validity

Discussion about conditions of validity.

Each condition of validity must be specified the more accurately possible to make easier the reproduction of axiom's experimental validation. Each modification of one of the conditions of validity must systematically imply a reproduction of formal experiment described in Section 3.8. In the context where all the conditions of table 3 are respected, the response time as a measure of time behavior is valid. In the opposite, i.e. if there is at least one condition which is not satisfied, the measure of response time is not valid. So, before any use of a measure, an external validation must be done. By external validation we mean a validation that enlarges the validity field of a measure. That means that at least one condition of validity has been modified and that experimental validation has been realized in the new context.

The axioms to satisfy are the preservation of relations order (5), the null value (6) and the non-negativity (7). However, the axioms of weak order (strongly complete and transitive) only concern the set A and the set of reals. The validity condition does not impact the respect of the axiom of weak order of the set of real. If conditions are changed, the set of reals *Re* will preserve the property of weak order and the transitivity. The same can be affirmed concerning the set A which contains the paradigms of the n-tiers Internet architecture. But, the other axioms (5,6,7) can be violated if the validity conditions are modified.

The number of active processes (the CPU load) is one of the validity conditions that has a strong impact on the validity of the response time. To verify this impact on the axioms (5),(6) and (7), an experiment could be the verification of a suitable enhancement of the CPU load. A suitable enhancement means that the CPU load is enhanced by using a process often active on a server. The launch of an office application is nonsense because this kind of program is not often active on a server. But, the launch of a database server would be a more appropriate approach.

4. Experimental validation of the homomorphism

4.1. Definition

Object of the study: the response time as a measure of the time behavior in terms of efficiency.

Purpose: The purpose is to validate the response time as a measure of the time behavior in terms of efficiency.

Quality focus: The quality focus is the validity of the response time for measuring the time behavior in terms of efficiency.

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

Context: The experiment is run in one single computer (PPC G3 300 Mhz, 160 Ram) using a software tool for measuring the response time of 8 programs. The table 4 gives the conditions of the measurement.

Processor	PPC 300 MHz G3 L2 Cache 512K
RAM	160 MB SDRAM
Hard disk	Toshiba 3,2 Go
OS	Mac OS 10.2.3
Network connectivity	None
Programming language	PHP 4.1.2
Apache Version	1.3.26
Browser	Lynx Version 2.8.4pre.2
Developer maturity	5 years of experience
Active process (CPU load)	Apache, Lynx, Shell
Number of connected clients	1
Request number	1

Table 4: Validity conditions of the experiment

4.2. Planning

Context selection: The context of the experiment is the software quality lab of the university of Namur, and hence the experiment is run off-line (not industrial environment). The experiment is specific since it focuses on the validity of the response time of a PHP application under Internet n-tiers architecture.

. It addresses a real problem, i.e. the validity of the response time measure.

4.3. Hypothesis formulation

Null hypothesis, H_0: the response time (defined in the standard ISO/IEC 9216) is not a valid measure (does not satisfy the axiom (5) of the empirical order preservation) of the time behavior in terms of efficiency for an Internet n-tiers architecture PHP application. There is no correlation (measured as rho, the Spearman correlation coefficient) between the empirical order of the 8 programs and the response time of theses programs. This idea can be expressed as:

H_0: rho = 0 with a level of significance $\alpha = 5\%$.

Alternative hypothesis, H_1: rho > $r(\alpha, n)$ where $r(\alpha, n)$ is the theoretical value (see Spearman statistical table).

4.4. Variables selection

The independent variables are the validity conditions of the table 4 and the dependent variable is the response time (measured in microseconds).

4.5. Experiment design

Blocking: The number of active process (CPU Load > 0%), the number of connected users, the number of requests to the web server, the operating system and all the validity conditions expressed in table 4 can affect the response time in a way that is not interesting for the scope of the current experiment. So, it is interesting to block these factors.

Balancing: The experiment uses a balanced design, which means that there is the same number of data (1000 response times) for each program.

Standard design type: The experiment design is a paired comparison design of type "one factor with two treatments". The factor is the order of the programs based on the efficiency and the two treatments are the empirical order and the numerical order.

4.6. Instrumentation

The measure of the response time is performed through a software timer. The timer makes the difference between two timestamps: the first one is the time before the execution of the program to measure and the second is the time after the execution. The timer has a precision of about a microsecond. The documentation of the timer is provided through the online documentation of the PHP native function microtime() [PHPDOC].

4.7. Validity evaluation

Internal validity is focused on *the relationship (...) observed between the treatment and the outcome, we must be sure (...) that it is not a result of a factor of which we have no control* [WOHL00]. The result cannot be

generalized outside the scope of this study. There is a large number of tests (equals to the number of measures per program), which ensures a good internal validity.

Concerning the *construct validity*, the experiment tries to prove that the response time is a good measure of the time behavior in terms of efficiency. The study is concerned with the relationship between theory and observation. The construct validity is not considered to be critical.

The *conclusion validity* of the experiment is not a problem. This validity is concerned with the relationship between the treatment and the outcome. This is what the study aims to prove: the correlation between the empirical order and the numerical order.

4.8. Operation

Preparation

The subjects (experts) are informed that the experiment goal is to validate the response time. The number of subjects is 2. The subjects must be experts in the Internet development.

The 8 programs are ready before the experiment is executed. The subjects can read the programs or execute them (without measuring) for estimating the time behavior. They must fill one single form where an ascendant order sorting of the 8 programs in terms of time behavior is asked. The programs are run on the same computer (see table 4 for the computer specifications) as the measurement.

The measurement tool (timer) is plugged in the programs to measure. The subjects do not receive the programs with the timer.

Execution

The experiment is executed just one time. The subjects must estimate the time behavior by consensus. Then, the measure of the response time is executed with the timer. Each program is measured 1000 times in just one shoot.

Data validation

It is assumed that the distribution of the variable response time is a normal distribution. The test of the normality hypothesis has been done graphically. The average of the response time is a good estimation of the mathematical expectation due to the large sample (n= 1000).

The distribution of the response time of the empty program (A_{11}) presents a bimodal distribution (see figure 1). The CPU load of Lynx text-based browser is lesser than the CPU load of MS Internet Explorer. So using Lynx instead of MIE allows having a normal distribution (see figure 2)

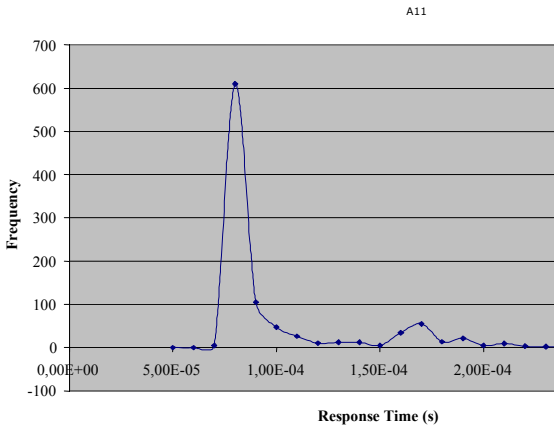

Figure 1: Script A1 with MS IE

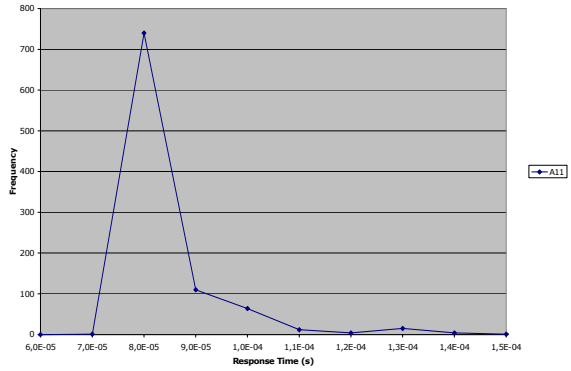

Figure 2 : Script A1 with Lynx browser

The average of the response time of the script A_{11} is not null (see table 6) but close to zero. The experimental zero is 8.33E-05 seconds and this is a standard. The standard zero is used for calibrating the measurement tool, i.e. the timer. The standard zero allows satisfying the axiom of nullity (6).

Script	Response Time (s)
A11	8.33E-05
A2	1.35E-04
A3	1.07E-04
A4	1.52E-04
A5	9.42E-03
A6	9.45E-03
A7	1.19E-01
A8	5.59E-01

Table 6: Response time average

The non-negativity axiom is satisfied. None of the values collected is less than zero. The hypothesis of normality distribution and the average as a good

estimation of the mathematical expectation allow us to assume that a negative value of the response time is not possible in this context.

The distributions of the programs A2, A3, A4, A5, A6 are normal distribution (see figure 3).

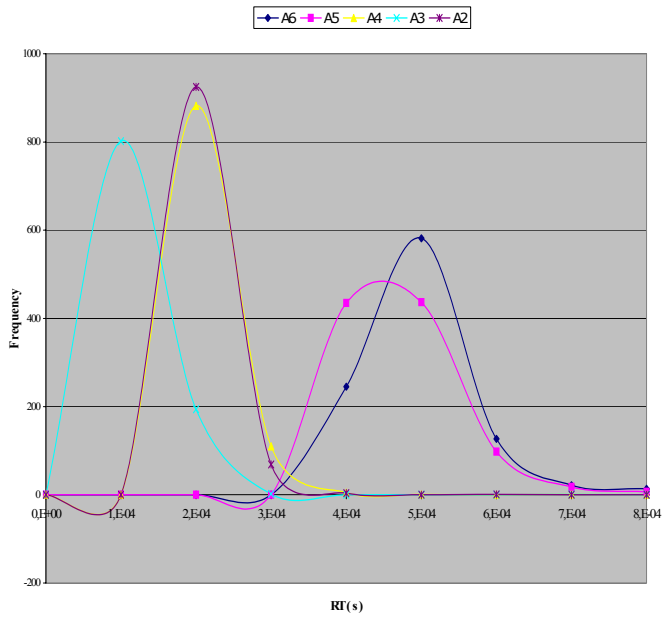

Figure 3: Distribution of the response time (RT)

The distributions of the programs A7, A8 are normal distributions (see figure 4).

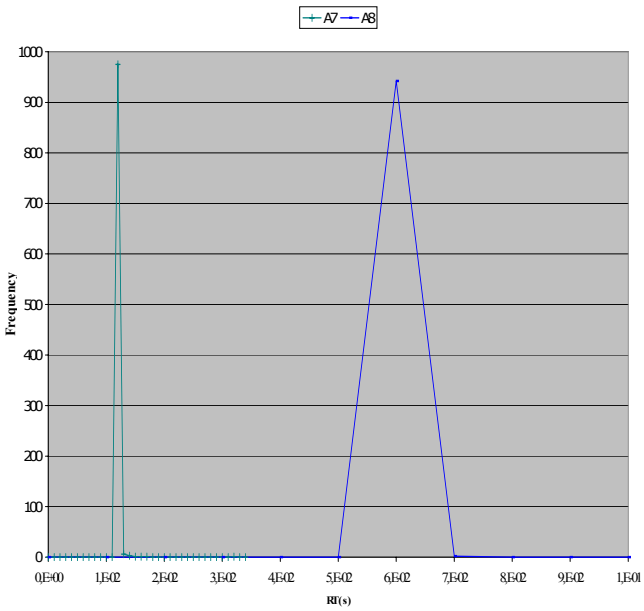

Figure 4: Distribution of the response time

4.9. Analysis and interpretation

Descriptive statistics

Table 7 shows the estimation order as given by the experts (X) and the ranking of the 8 programs based on the response time average (Y). The colon d_i is a term of the Spearman coefficient (rho) and represents the difference between X_i and Y_i (i is the rank).

	X	Y	d_i	$d_i * d_i$
A_{11}	1	1	0	0
A_2	2	3	-1	1
A_3	3	2	1	1
A_4	4	4	0	0
A_5	5	5	0	0
A_6	6	6	0	0
A_7	7	7	0	0
A_8	8	8	0	0

Table 7 : Ranking

The Spearman coefficient equals 0.943 and the theoretical value at a level of significance $\alpha = 5\%$ equals 0.829. So, the null hypothesis is rejected at a level of significance $\alpha = 5\%$.

The theoretical value at $\alpha = 2.5\%$ equals 0.886. So, the null hypothesis is also rejected at $\alpha = 2.5\%$.

But, the theoretical value at $\alpha = 1\%$ equals 0.943. The null hypothesis must be accepted at a level of significance of $\alpha = 1\%$.

It exists a correlation between the estimation of the time behavior done by the experts and the measured response time. The probability that this correlation would occur by chance equals 2.5%.

5. Conclusion

The validation process presented here is build in 7 steps:

1. Definition of the objectives of the measure
2. Selection of the experts
3. Specification of an empirical relational system
4. Specification of a numerical relational system
5. Mathematical expression of the homomorphism
6. Validation of the empirical system
7. Validation of the numerical system

The steps 6 and 7 can be done in a experimental or theoretical way. The validation of the weak order axiom for the set of programs is a theoretical validation. But, the validation of the homomorphism axiom must be done via a formal experience.

The objective of the measure determines the relational systems (the empirical and the numerical). It specifies the type of the empirical objects, the relations between them and the operations. In the present paper, the specified relation is "faster than". The objective is to compare different programs in terms of efficiency. In this case, it is not necessary to define operations on the programs. However, if the measure goal is to compare in terms of efficiency programs coming from the concatenation of other programs, the concatenation operation should be defined in the relational systems (empirical and numerical) with supplementary axioms to prove. The kind of question is: "Is the program $P_3 = P_1 + P_2$ faster than P_1 or P_2 ?" Such problems must be solved by experts who must specify, by consensus, these properties in a formal way. This will be the object of future works.

The problem of the experts is a confidence problem. How to know whether the experts knowledge is relevant and exhaustive? The international framework provided by the ISO represents a suitable structure for reaching a consensus and for having the benefit of an acknowledgement.

The paradigms are also a critical point in the validation process. The choice of the paradigms is done during the specification of the relational empirical system (step 3). The paradigms must reflect representative situations of the domain, the technology (client-server) and the measure goal (software attribute, relations, operations). The problem of the efficiency is a critical point of the n-tiers architecture. In this architecture, the connections to a data source, the session handling and other patterns must be taken into account for specifying the paradigms. This is true for any measure to be validated. Experts must elaborate patterns for each domain, technology or attribute. The paradigms become an instance of the previous patterns. This approach ensures the relevance and the completeness of the paradigms.

It is assumed that the validity conditions impact the measure validity. The correlation between the validity conditions and the measure validity must be verified in an experimental way. It is also important to know how each condition influences separately the measure validity. This experimental validation can often be hard to do. For example, how to test the correlation between the measure validity and Microsoft Windows 2000 ? For doing that, a change of processor must be done and in this case two of the validity conditions are modified at the same time. So, it is impossible to measure the operating system impact on the validity without modifying the processor. The test of correlation has not been done in this paper because it should be out of the scope. It will be the object of future works.

A measure is valid if it satisfies the properties (axioms) specified by experts. What is not expressed in a mathematical way (measurement theory) is out of the scope of the validation process. The measurement theory is a framework, which reduces the ambiguity of the expression *measure validity* and enhances the operability of the validation process.

This paper validates the response time as measure of the time behavior regarding the software efficiency of a n-tiers Internet application. The software practitioners can use the response time as defined above if and only if all the validity conditions are satisfied.

6. References

[BIEM92] Bieman, J.M.; Schultz, J.
« An Empirical Evaluation (and Specification) of the all-du-paths. Testing Criterion. », Software Engineering Journal, Volume 7, No. 1, pp. 43-51, January 1992.

[FENT96] Norman E. Fenton, Shari L. Pfleeger
« Software Metrics : A Rigorous Approach », Chapman & Hall, 1996

[HEND96] Henderson-Sellers
« Object-Oriented Metrics – Measures of Complexity », Prentice Hall, 1996

[IEEE93] IEEE Computer Society
« IEEE Standard for a Software Quality Metrics Methodology. », IEEE Standard 1061

[ISO9126] Software Engineering-Product Quality
« Part 1: Quality Model», ISO/IEC TR 9126-1, 1999.
« Part 2: External Metrics», ISO/IEC TR 9126-1, 1999.

[JACQ99] Jean-Philippe Jacquet, Alain Abran, Robert Dupuis
« Une analyse structurée des méthodes de validation de

métriques », , 1999
[PHPDOC]
http://www.php.net/manual/en/function.microtime.php
[ROBE79] Fred S. Roberts
“Measurement Theory with Applications to Decisionmaking, Utility, and the Social Sciences”, Encyclopedia of Mathematics and its Applications Addison Wesley Publishing Company, 1979
[WOHL00] Claes Wohlin *et al.*
« Experimentation in software engineering : An introduction », Kluwer Academic Publisher, 2000
[ZUSE99] Horst Zuse
« Validation of measures and prediction models », International Woorkshop o Software Measurement, 1999