Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

CONSTRUCTION OF A SYSTEMIC QUALITY MODEL FOR EVALUATING A SOFTWARE PRODUCT

MARYOLY ORTEGA [*]**, MARÍA A. PÉREZ** [**] **and TERESITA ROJAS** [**]

Abstract. Quality is currently considered one of the main assets with which a firm can enhance its competitive global position. This is one reason why quality has become essential for ensuring that a company's products and processes meet customers' needs. A recent innovation in the systems area is the development of a set of mechanisms and models for evaluating quality. This article describes the design of a Quality Model with a Systemic Approach to Software Products that assesses a product's efficiency and effectiveness. Different quality models were studied: McCall, Boehm, FURPS, ISO 9126, Dromey, ISO 15504 in an attempt to identify the aspects present in these models that are deemed important in a Systemic Quality model. We designed a model prototype that reflects the essential attributes of quality. This model was evaluated using a method so it can be validated and also enhanced. The evaluation method consisted of: designing a survey, formulating, validating and applying the measurement instruments; defining an algorithm to obtain the quality estimate and analyzing the results. The model prototype enabled the strengths and weaknesses of the software products studied to be identified. When evaluating a software product using the model prototype, it was possible to ascertain its compliance with the standards and use the results to improve it. Since the evaluation was systemic, processes that affect certain characteristics of the product could be identified. Companies can benefit from the model proposed because it serves as a benchmark that allows their products to evolve and be competitive.

Keywords: Software Product Quality, Quality Model, Systemic Quality, Metrics, ISO 9126, Dromey's Model.

* *Laboratorio de Computación, Universidad Ezequiel Zamora, Guanare. Edo. Portuguesa 3310, Venezuela.* **mortega@reacciun.ve**

***Departamento de Procesos y Sistemas – LISI, Universidad Simón Bolívar, Caracas 89000, Venezuela.* **movalles@usb.ve**; **trojas@usb.ve**

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

1. Introduction

Certification is important to avoid purchasing software of questionable quality. A reasonable starting point for software certification might be to consider one of the existing approaches. These include certifying developers in order to demonstrate a set of specific skills, assessing the behavior of software products and/or certifying that processes are properly followed. Voas (Voas 1999) refers to these approaches as the triangle of software quality certification, covering processes, products and personnel.

Callaos and Callaos (Callaos and Callaos 1996) also stress the importance of this balance, referring to the Product-Process aspects of a system.

In their proposal Callaos and Callaos link the elements of the series of local qualities identified, in order to obtain a "systemic quality" rather than just combining them, as in the case of the majority of total quality methodologies (Callaos and Callaos 1996). According to them, the meaning of design includes: design as a product and design as a process. The system designed, the product, differs from the system of human activities, the process, through which the product is being designed. Hence four types of quality are identified: product efficiency, product effectiveness, process efficiency and process effectiveness (Callaos and Callaos 1996).

Pfleeger and Humphrey point out that certifications, standards and measurements must take into account a balanced proportion of processes, products and resource requirements (Pfleeger 1998; Humphrey 1989), aware that processes are groups of activities related to software; products are artifacts, documents resulting from a process and resources are entities required by a process (Fenton and Pfleeger 1997).

Recent evidence shows that the software industry is starting to show an interest in obtaining formal certification. For example, Linuxcare is setting up laboratories to certify the hardware and software to be used in the Linux operating system. Other companies that certify processes, personnel and products are: Det Norske Veritas, (DNV, Oslo), International Computer Security Association, KeyLabs, Technical Inspection Association from the German Technischen Uberwachungverein (TUV, Germany) and Verisign (Voas 1999). Besides, large lists of software testing services are published in Internet: Compuware Corporation, VeryCode Inc., NTS - Hardware & Software Testing Labs are some of them (Google, 2003).

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

On a worldwide scale the outlook is quite different, as can be seen from the number of software quality conferences publicized on the Internet and in specialized magazines.

Generally speaking, software product certification is based on models, which is why an in-depth look at this aspect is warranted.

A model is an abstract form of reality, enabling details to be eliminated and an entity or concept to be viewed from a particular perspective (Fenton and Pfleeger 1997). There are all kinds of models: cost estimation models, quality models, maturity models, etc. Models can be presented in different ways, such as in the form of equations, functions or diagrams. This makes it possible to show how components are related, so they can be examined, relationships understood and opinions formed (Fenton and Pfleeger 1997).

The literature on the subject covers several product quality models, the best known in chronological order of appearance are: McCall (McCall et al. 1977), Boehm (Boehm et al., 1978), FURPS (Grady and Caswell, 1987), ISO 9126 (ISO/IEC 9126, 1991) and Dromey (Dromey, 1996). These were analyzed in order to obtain relevant elements for this research. The importance of this study lies in the growing interest among systems development organizations in obtaining product certification, as well as the lack of mechanisms for doing so.

The fundamental axiom of product quality is that the tangible internal characteristics or properties of a product determine its external quality attributes. Developers must build these internal properties into the product to demonstrate the external quality attributes desired (Dromey, 1996).

The objective of this study is to identify the product quality characteristics needed to obtain Systemic Quality. Thus product efficiency and effectiveness are considered, without forgetting that this is an integral part of the Systemic Quality concept, bearing in mind that this approach must be part of the Process Quality model.

The article is therefore structured as follows: second section gives the background, third section describes the method used, fourth the formulation of the model, fifth the design of the survey and case study, sixth shows how the model is applied, seventh the results analysis and eighth the limitations of the empirical study. The conclusions and recommendations are presented in the ninth and last section.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

2. Background

The literature on the subject contains a significant number of quality models that have gradually evolved. By analyzing them it was possible to identify the principal elements to design a new Software Quality Model with a Systemic Approach. The quality aspects considered in them are described below.

2.1. The McCall Model

This model pinpoints three working areas in systems: product operation, product revision and product transition (McCall et al. 1977). Product Operation refers to the product's ability to be quickly understood, efficiently operated and capable of providing the results required by the user; the following product characteristics are taken into consideration: Modifiability, Reliability, Efficiency, Integrity, Usability. Product Revision is related to error correction and system adaptation, this aspect is important because it is generally considered the costliest part of software development; the following characteristics are covered: Maintainability, Flexibility and Testability. Product Transition may not be important in all applications, although the trend towards distributed processing, together with rapidly changing hardware, is likely to increase its importance; Desirable product characteristics in this area of work are: Portability, Reusability and Interoperability.

One of the major contributions of the McCall model is the relationship created between quality characteristics and metrics, although there has been criticism that not all metrics are objective. One aspect not considered directly by this model was the functionality of the software product.

2.2. The Boehm Model

Upon analyzing the Boehm model one sees that it begins with the software's general utility. It is thereby affirming that, over and above anything else, a Software System must be useful, otherwise its development will have been a waste of time, money and effort. It looks at utility from various dimensions, considering the types of user expected to work with the system once it is delivered. General utility is broken down into Portability, Utility and Maintainability. Utility is further broken down into Reliability, Efficiency and Human Engineering. Maintainability is in turn broken down into Testability, Understandability and Modifiability (Boehm et al. 1978).

The Boehm model is similar to the McCall model in that it represents a hierarchical structure of characteristics, each of which contributes to total quality. Boehm's notion includes users' needs, as McCall's does; however, it also adds the hardware yield characteristics not encountered in the McCall model.

2.3. The FURPS Model

The FURPS model takes into account the five characteristics that make up its name: Functionality, Usability, Reliability, Performance, and Supportability. When the FURPS model is used, two steps are considered: setting priorities and defining quality attributes that can be measured. Grady and Caswell (Grady and Caswell 1987) note that setting priorities is important given the implicit trade-off, i.e. one quality characteristic can be obtained at the expense of another. One disadvantage of this model is that it fails to take account of the software product's portability.

2.4. ISO 9126

ISO 9126 defines product quality as a set of product characteristics. The characteristics that govern how the product works in its environment are called external quality characteristics, which include, for instance, Usability and Reliability. The characteristics relating to how the product was developed are called internal quality characteristics; they include, for instance, size, tests and failure rate, exchange rate, structure, etc. taken in the development of the product (ISO/IEC 9126-1.2 1998).

ISO 9126 indicates that some component of the software quality must be described in terms of one or several of six characteristics: Functionality, Usability, Maintainability, Reliability, Portability and Efficiency. In turn, one of these six characteristics is defined as a set of attributes that are supported by a relevant aspect of the software; each one can be refined at multiple levels of subcharacteristics (ISO/IEC 9126-1.2 1998).

The internal attributes of the software influence the external attributes; thus there are internal aspects and external aspects for the majority of the characteristics.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

In the ISO 9126 documents, the quality characteristics are defined with the associated subcharacteristics (ISO/IEC 9126-1.2 1998):

- Functionality: Suitability, Accuracy, Interoperability, Compliance, Security
- Reliability: Maturity, Fault Tolerance, Recoverability
- Usability: Understandability, Learnability, Operability
- Efficiency: Time behavior, Resource behavior.
- Maintainability: Analyzability, Changeability, Stability, Testability
- Portability**:** Adaptability, Installability, Conformance, Replaceability

One of the advantages of this model is that it identifies the internal characteristics and external quality characteristics of a software product. However, at the same time it has the disadvantage of not showing very clearly how these aspects can be measured.

2.5. The Dromey Model

Dromey proposes a working framework for building and using a practical quality model to evaluate Requirement Determination, Design and Implementation phases. This information can be used directly to build, compare and evaluate better quality software products (Dromey 1996).

In referring to the well-known expression "build quality into software", Dromey points out that high-level quality attributes, such as Reliability and Maintainability, cannot be built into the software. What can be done though is to identify a set of properties (such as modules without side effects) and build them up consistently, harmoniously and fully to provide reliability and maintainability. Links must be forged between the tangible properties of the product and the high-level quality attributes (Dromey 1996).

Dromey proposes three models, depending on the products resulting from each stage of the development process: requirements model, design model, and implementation quality model (programming).

In comparing this model to ISO 9126, additional characteristics like Process-Maturity and Reusability are noticeable. It is important to point out the weighting Dromey gives to Process Maturity, an aspect not considered in previous models.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

This model seeks to increase understanding of the relationship between the attributes (characteristics) and the subattributes (subcharacteristics) of quality. It also attempts to pinpoint the properties of the software product that affect the attributes of quality.

After analyzing various product quality models, the different quality attributes or characteristics found in each of them must be compared, see Table 1.

Table 1 shows that the quality characteristics found in the majority of the models are: Efficiency, Reliability, Maintainability, Portability, Usability and Functionality, which have been present in more recent models. Because they are present in all the models studied, they can be considered essential and worthy of study.

Table 1. Quality characteristics present in the different models: Boehm, McCall, FURPS, ISO 9126, Dromey

Quality Characteristics	Boehm	McCall	FURPS	ISO 9126	Dromey
Testability	x	x		x	
Correctness		x			
Efficiency	x	x	x	x	x
Understandability	x			x	
Reliability	x	x	x	x	x
Flexibility		x	x (extensibility, adaptability, maintainability)		
Functionality			x	x	x
Human Engineering	x				
Integrity		x		x(Security)	
Interoperability		x		x(Functionality)	
Process Maturity					x
Maintainability	x	x	x	x	x
Changeability	x				
Portability	x	x		x	x
Reusability		x			x

Other studies that support the idea of systemic quality, taking into account the quality of the product and the process, have been presented by Dromey (Dromey 1996) and Voas (Voas 1999). Dromey says there no quality process can exist if it is not based on a product quality model.

2.6. Systemic Quality Model

A model that differs slightly from those previously studied, since it has no hierarchical structure, is the Callaos model (Callaos and Callaos 1996). There is a similarity between the concepts used by Callaos as regards product efficiency and effectiveness and the definitions of the product's internal and external characteristics.

What Callaos means is that there must be a balance between efficiency and effectiveness, which leads us to conclude that special care must be taken with the product's internal and external characteristics. Systemic thinking, because of its inherent nature, calls for interdependence and cooperation. Callaos and Callaos identify the following relationships as the basis for attaining Global Systemic Quality (Callaos and Callaos 1996):

- Product-Process
- Efficiency-Effectiveness
- User-Customer

Figure 1 shows the relationship between these concepts.

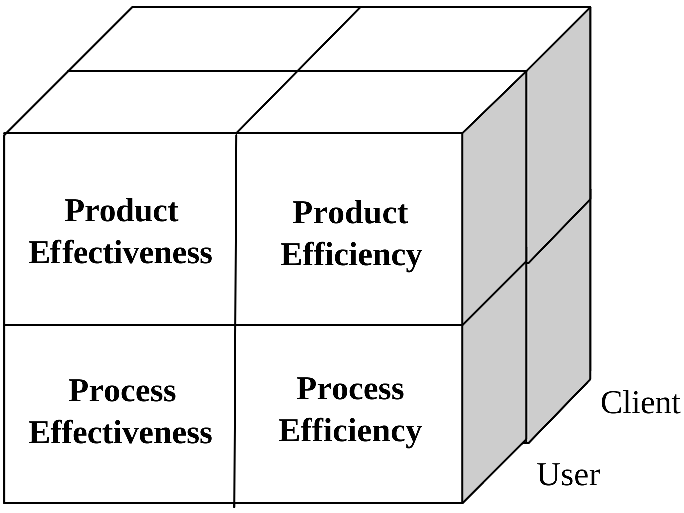

Figure 1. Global Systemic Quality Matrix. Source: Adapted (Callaos and Callaos 1996)

These partial qualities are related to each another and must be integrated within a Systemic Global Quality System design. Global Quality is not the sum of its partial qualities. Global Quality implies a commitment among the entire series of qualities that leads to a global optimum with some sacrifice of the partial optimum (Callaos and Callaos 1996). That is what makes it systemic.

Product efficiency and effectiveness are differentiated from one another.

Product Efficiency is determined by internal design and programming activities since an efficient product is obtained when correct physical design and programming practices are applied (Rojas and Pérez 1995).

Product Effectiveness is determined by activities involving requirement identification, interface design and general network design (point location), since this is closely related to compliance and user comfort (Rojas and Pérez 1995).

Just as in the product, the process too has effectiveness and efficiency dimensions.

Process Efficiency is associated with project management activities which include meeting deadlines, increasing productivity and saving resources (Rojas and Pérez 1995).

Process Effectiveness is related to general management activities, such as leadership, change management, human and group relations, as these lead to good relations between the members of the team responsible for developing Information Systems (Rojas and Pérez 1995).

These four qualities differ if we consider the customer and the user of the design, so eight qualities involved in obtaining Systemic Global Quality would be obtained.

The objective of this work is to focus on product quality. This is why product efficiency and effectiveness are considered, not forgetting that this can be included in the concept of Systemic Global Quality.

3. Method

The method followed in this study comprises the following steps:

a. Documentary and bibliographical research to create a theoretical framework of reference.

b. Formulation of the product quality model. This involved:

   - Identifying the high level quality characteristics for the software product;

- Creating a taxonomy of the most significant and tangible characteristics provided by Systemic Quality;
- Proposing a set of metrics corresponding to the four dimensions of Systemic Quality in order to link the characteristics identified in the previous step.

For the purpose of this research, the metrics were proposed taking into account the point of view of an organization that evaluates software.

c. Design of the survey for evaluating the prototype model. The sample must meet certain requisites so that the information necessary for evaluating the product may be obtained.
   - The products must have a well-defined user (market), which will make it easier to evaluate its requirements.
   - The installations must facilitate the evaluation.
   - The product must have been on the market for some time (2 years), so that users with experience of it can be identified.
   - Technical support must be available for product evaluation.
   - The company must understand the need to evaluate its Systems.
   - It must be easy to access and evaluate the different products of the various stages of the System's development process (requirements, design and code).
   - The quality model prototype must be applied by two evaluators to two software products, so it can be refined.

   Design of the survey and the case study are dealt with in Section 5.

d. Identification, formulation and validation of the measurement instrument. A measurement instrument in the case of software product evaluation can be evaluated in different ways. At this stage the most suitable instrument for the evaluator to use to capture the data associated with the product's quality characteristics (efficiency and effectiveness) is chosen and justified. The user must be interviewed so information on the product's effectiveness and other details the evaluator deems pertinent can be obtained. In addition to this, the quality characteristics of the evaluation and the different types of reports that would be used in it must be identified. This step is dealt with in Section 6.1.

e. Application of the instrument to two software products selected when determining the sample and refining the model. During this step, the stages in the evaluation process and the resources used in it had to be defined.

f. Application of the algorithm for obtaining the quality estimate. During this step the algorithm to be used to obtain the partial or total estimate of the quality of the software product is identified.

g. Analysis of the results obtained. Once the data and final results of the evaluation have been obtained, they are analyzed and relevant conclusions on the behavior of the model are reached. This analysis can be used to determine how well the model reflects the quality of a given product, and at the same time make it possible to refine:

- the results of applying the model;
- the relationship between product efficiency metrics and product effectiveness metrics, as well as the properties of the processes;
- the developer's recommendations regarding the software's strengths and weaknesses;
- the results will serve as the basis for the model's automation, which will facilitate its use and dissemination.

4. Formulation of the model

This section presents the formulation of the Software Product Quality Model prototype based on the Systemic Quality (Callaos and Callaos 1996). The model proposed takes the following components into account:

- Product efficiency and effectiveness as partial quality of the Systemic Quality model.
- The quality characteristics of the Dromey model and ISO 9126: Efficiency, Reliability, Functionality, Maintainability, Portability and Usability.
- The relationship used in the McCall model between the quality characteristics and the metrics.
- The metrics existing in the literature on the subject will be studied, and an attempt made to relate them to the quality characteristics will be considered in the model.

Product Effectiveness will be represented by ISO 9126, specifically as regards the external characteristics defined, which take into account how the product beha ves in its environment: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

Next comes the Product Efficiency considered in the Systemic Quality Model and the Dromey model, which takes into account the product's internal characteristics or properties, including: properties of the requirements, properties of the design and properties of the implementation, represented in the Dromey model.

Lastly, Process Effectiveness and Process Efficiency are essential elements of the Systemic Quality Model, but they are not present in the Dromey model or in ISO 9126. This is why other models must be identified to complement this product's models with aspects of the process. There are different process quality models, in particular ISO 15504 (ISO/IEC TR 15504-2 1998), for evaluating software development processes. However, only the processes covered by ISO 15504 which favor software quality characteristics will be considered within the framework of this research. The resulting model is shown in Figure 2.

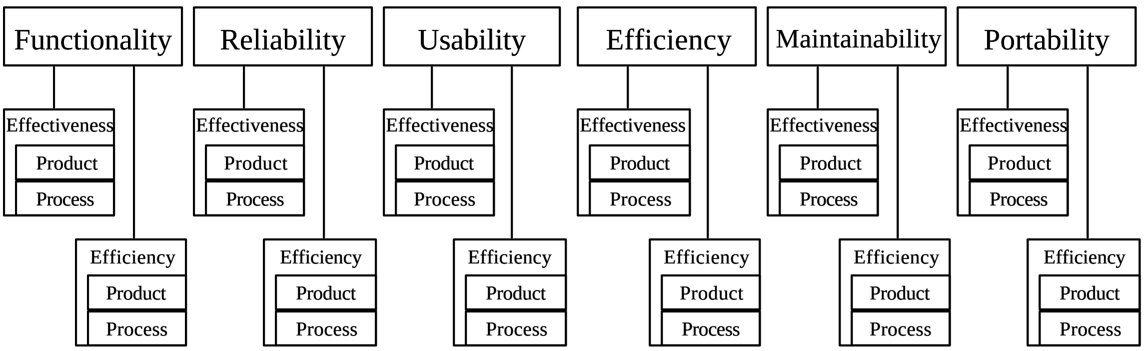

Figure 2. Elements of the Systemic Quality model for Software Products

Each characteristic is described below, indicating the subcharacteristics associated with each of the dimensions that interact to obtain systemic quality

Functionality is the ability of a software product to provide functions that meet specific and implicit needs when software is used under specific conditions. Functionality takes into account adaptation to the purposes, precision, interoperability and security of the software product. Table 2 shows the subcharacteristics associated with the Functionality, grouped by dimension.

Table 2. Subcharacteristics of Functionality in Systemic Quality dimensions

FUNCTIONALITY		
Dimension	Product Effectiveness	Product Efficiency
Product	Suitability, Accuracy, Interoperability, Security	Correctness, Structured, Encapsulated, Specified
Process	Acquisition preparation, Supply, Construction, Testing, Validation, Audit, Quality Management	Supply, Audit., Quality Management

Reliability is the capacity of a software product to maintain a specified level of performance when used under specific conditions. Table 3 shows the subcharacteristics associated with Reliability, grouped by dimension.

Table 3. Subcharacteristics of Reliability in Systemic Quality dimensions

RELIABILITY		
Dimension	Product Effectiveness	Product Efficiency
Product	Maturity, Fault Tolerance, Recoverability	Correctness, Structured, Encapsulated
Process	Audit, Quality Management	Operational Use, Integration and testing, Maintenance, Audit, Quality Management

Usability is the capacity of a software product to be attractive, understood, learned and used by the user under certain specific conditions. Table 4 shows the subcharacteristics associated with Usability, grouped by dimension.

Table 4. Subcharacteristics of Usability in Systemic Quality dimensions

USABILITY		
Dimension	Product Effectiveness	Product Efficiency
Product	Understandability, Learnability, Attractiveness, Operability	Complete, Consistent, Effective, Specified, Documented, Self-Descriptive
Process	Customer Support, Documentation, Audit, Quality Management	Operation, Customer Support, Audit, Quality Management

Efficiency is the capacity of the software product to perform adequately under specific conditions, depending on the amount of resources used. Table 5 shows the subcharacteristics associated with Efficiency, grouped by dimension.

Table 5. Subcharacteristics of Efficiency in Systemic Quality dimensions

EFFICIENCY		
Dimension	Product Effectiveness	Product Efficiency
Product	Time Behavior, Resource Behavior	Effective, Non-redundant, Direct, Utilized
Process	Audit, Quality Management	Audit, Quality Management

Maintainability is the capacity of the software to be modified. Modifications can include corrections, improvements or adaptations of the software to adjust to changes in the environment, in terms of functional requirements and specifications. Table 6 shows the subcharacteristics associated with Maintainability, grouped by dimension.

Table 6. Subcharacteristics of Maintainability in Systemic Quality dimensions

MAINTAINABILITY		
Dimension	Product Effectiveness	Product Efficiency
Product	Analyzability, Changeability, Stability, Testability	Coupled, Cohesive, Encapsulated, Maturity Attributes, Control Structure, Information Structure System, Descriptive, Correctness, Structural
Process	Requirement Elicitation, Analysis and Design, Documentation, Audit, Quality Management	Requirement Elicitation, Analysis and Design, Integration and Testing, Audit, Quality Management

Portability is the capacity of the software product to be transferred from one environment to another. Table 7 shows the subcharacteristics associated with Portability, grouped by dimension.

Table 7. Subcharacteristics of Portability in Systemic Quality dimensions

PORTABILITY		
Dimension	Product Effectiveness	Product Efficiency
Product	Adaptability, Installability, Co-existence, Replaceability	Consistent, Parameterized, Encapsulated, Cohesive, Specified, Documented, Self-descriptive, Non-redundant
Process	Software Requirement Analysis, Audit, Quality Management	Software Requirement Analysis, Audit, Quality Management

Having defined the subcharacteristics associated with each quality dimension, the metrics associated with each subcharacteristic were listed. In order to obtain quantitative values individually and as a whole, an estimate of the quality of each critical subcharacteristic is obtained calculating the percentage of the metrics that fall within the optimum values according to the user requirements (equal to or greater than 3). Besides, the median of the metrics for each subcharacteristic is used as a measure of central tendency. In order to obtain the link between the product and process characteristic, each quality subcharacteristic related to the product has metrics associated with the process subcharacteristics.

Table 8 shows the metrics associated with the Accuracy subcharacteristic of the Functionality characteristic. The metrics adapted from ISO/IEC 9126 are related to the product's own characteristics, and the metrics adapted from ISO/IEC TR 15504-2 measure the presence of processes that support product quality.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

Table 8. Metrics for measuring Accuracy

Metric	Inf	Sup	Description
Incomplete results *	1	5	1= All; 2= Nearly all; 3= Few; 4= Very few; 5= None
Incorrect results *	1	5	1= All; 2= Nearly all; 3= Few; 4= Very few; 5= None
Unexpected results issued *	1	5	1= All; 2= Nearly all; 3= Few; 4= Very few; 5= None
Test plans **	1	5	5= Completely defined; 4= Nearly all defined; 3= Semi defined; 2= Little defined; 1= Not defined
Test frequency**	1	5	5= Daily; 4= Weekly; 3= Monthly 2= Half-yearly; 1= Yearly
Validation criteria **	1	5	5= Completely defined; 4= Nearly all defined; 3= Semi defined; 2= Little defined; 1= Not defined
Validation activities **	1	5	5= Completely defined; 4= Nearly all defined; 3= Semi defined; 2= Little defined; 1= Not defined
Problem solving **	1	5	5= Always; 4= Nearly always; 3= Sometimes; 2= Infrequently; 1= Never
Communication of the results of the validation **	1	5	5= Excellent; 4= Good; 3= Average; 2=Below average; 1= Unacceptable

* Adaptation (ISO/IEC 9126-1.2 1998)

** Adaptation(*ISO/IEC TR 15504-2 1998)*

It is important to point out here that there are two processes, Auditing and Quality Management, that affect each of the quality characteristics specified within the model. For example, the Auditing process has the following metrics (ISO/IEC TR 15504-2 1998): Has an auditing strategy been implemented? Satisfaction with the audit. Are there pre-determined points (dates, activities) where audits are carried out? Is there an audit strategy to determine whether products, services or processes comply with requirements, plans and contracts? Are audits conducted by an independent unit? Are problems detected during an audit communicated to those responsible for taking corrective action?

Once the model has been formulated and the characteristics, subcharacteristics and metrics are known, the bases of the experiment used to demonstrate the model prototype are explained.

5. Design of the survey and case study

The purpose of this section is to explain the design of the survey in detail. It covers the selection and estimation method used and the size of the sample (Seijas 1999). Also in this section the sample will be determined and its choice justified, as well as the characteristics of the sector where the evaluation is made.

5.1 Design of the Survey

The internal characteristics of a design influence the attributes of quality, and thereby total quality. It is therefore significant to compare products that have been developed for similar or related objectives. Thus, demonstration of the model's predictability calls for evaluation and comparison of several products designed for the same series of requirements. This evaluation of the total quality determined by the model needs to be in line with the total quality requirements or characteristics as perceived by analysts, developers and users.

Based on these objectives, the following approach was adopted to demonstrate the result of the total quality evaluation produced by the quality model.

- *Select a set of products for demonstrating the model*: Since the perception of quality varies widely and depends on the evaluator's point of view, a variety of evaluators capable of acting as factors in an evaluation must be used. Two average products are used in the demonstration of the model. The products used are geared towards the same requirements and are easily available for evaluation. The size of the projects reflected by the Function Points estimating method (number of user inputs, number of user outputs, number of user enquiries, number of files, number of external interfaces), made them ideal candidates for the model and the evaluators (Albrecht and Gaffney 1983; Pressman 1998).
- *Determination by the two evaluators*: A group of 2 evaluators was engaged to study the quality of the 2 products. The evaluators have spent 2 to 7 years in software development. They have the same education background. The study was conducted over a 3-month period.

- The evaluators (E1 and E2, see Table 9) were asked to assess two products (P1 and P2), including their requirements, design, implementation and documentation.
- The experiment was designed (see Table 9) to produce four results from the evaluations: C11, C12, C21 and C22, which were analyzed in section 7.

Table 9. Design of the experiment

	Product P1	Product P2
Evaluator E1	C11	C12
Evaluator E2	C21	C22

All the above is based on the following:

1. Two software products are required. They must be inter-related and work according to the same series of requirements, in order to guarantee that the model behaves in a stable manner when faced with similar circumstances and at the same time is able to show product differences.
2. Two evaluators guarantee that the model will behave the same way, regardless of the evaluator.

After designing the survey, the next step was to identify the sample to which the prototype of the model was applied. In the following section the selection of the sample and the characteristics of the sector to which they belong are justified.

5.2 A case study

The firm where the evaluation was carried out belongs to the banking sector and falls within the service company category. Its financial intermediation functions can be broken down into credit transactions and money receipt operations.

In this research, two Risk Management projects were selected. These are: "Special Monitoring" and "Appraisal".

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

The “Special Monitoring” and “Valuation” systems were chosen out of several options for the following reasons: the users are clearly defined; the following software product components are available: requirements, design and code; technical support is available to carry out the evaluation; the product has user acceptance; the two products have a common environment, they share the same environment and the same users; the systems interact and they have the same objectives and corporate priorities.

All this means that the evaluation can be undertaken in a stable environment, avoiding factors in the environment that might influence the results of the evaluation and not be able to be controlled by the model.

General descriptions of the products to be evaluated are given below:

a) “Special Monitoring” Product.

Special Monitoring is the name given to follow-up of risky clients where there are circumstances or warning signs liable to affect normal development and/or prompt payment of their transactions.

b) “Valuation” Product

The Valuation System is an internal management tool for the approximate measurement of the quality of a firm and the risk it may entail, with a view to establishing its current and future capacity to meet its payment commitments.

6. Application of the model

In order to complete the last four steps of the method described in Section 3; first of all the instruments to be used in the evaluation are identified and validated. Then the procedure used for applying the instrument is presented. Next the algorithm used to obtain the estimate of the quality is described. Lastly the analysis of the results of the evaluation is given.

6.1 Identification, Preparation and Validation of the Measurement Instrument

In this step the most suitable instrument for the evaluator to use to capture the data associated with the product's quality characteristics (efficiency and effectiveness) is selected and justified.

Users must be interviewed to obtain further information regarding its effectiveness. The quality requirements had to be identified in the evaluation, as well as the different types of report to be used in it.

Since product's effectiveness is measured in terms of how it works in its environment, questions had to be chosen for the users of the product.

The questions were classified according to quality characteristics and subcharacteristics and asked each person interviewed, depending on his/her role (Table 10). Users can only answer questions on *Functionality, Reliability and Usability.* In the Analysts' and/or Developers' questionnaire, the characteristics to be evaluated included: Functionality, Reliability, Usability, Maintainability, Efficiency and Portability. The questions put to the Project Leaders are to a large extent related to existing Product Software Development processes.

Table 10. Characteristics evaluated in the questionnaire depending on the interviewee's role

Characteristic	**Sub-characteristics**	**Interviewee's role**
Functionality	Suitability	User, Analyst/Developer, Project Leader
	Accuracy, Interoperability, Security	User, Analyst/Developer
	Correctness, Structured, Encapsulated, Specified	Analyst/Developer
Reliability	Maturity	User, Analyst/Developer, Project Leader
	Recoverability	User, Analyst/Developer
	Correctness, structured, encapsulated	Analyst/Developer
Usability	Understandability	User, Analyst/Developer
	Learnability, Operability	User, Analyst/Developer, Project Leader
	Complete, Consistent, Effective, Specified, Documented	Analyst/Developer
Efficiency	Time Behavior, Resource Behavior, Effective, Non-redundant, Direct, Utilized	Analyst/Developer

Maintainability	Analyzability, Stability ,Coupled, Cohesive, Encapsulated, Software Maturity attributes, Control Structure System, Information Structure System, Descriptive, Physical Attributes of Support Documentation, Correctness, Structural, Modularity, Specified, Documented, Self-descriptive	Analyst/Developer
	Changeability, Testability	Analyst/Developer Project Leader
Portability	Adaptability	Analyst/Developer Project Leader
	Installability, Co-existence, Changeability, Consistent, Parameterized, Encapsulated, Cohesive, Specified, Documented, Self-descriptive, Non-redundant	Analyst/Developer
Audit and Quality Management	Audit and Quality Management	Analyst/Developer, Project Leader

The method used to validate the measurement instrument was to check that there was no significant difference between evaluations of the same software product by two different evaluators.

For this research work, the measurement chosen was reliability. According to Hernández et al. (Hernández et al. 1998), in this procedure the same measurement instrument is applied at least twice to the same group of people, after a certain period. If the correlation between the results of the different applications is highly positive, the instrument is considered to be reliable.

6.2 Application of the Instrument to two Software Products and Refinement of the Model

a. First of all several surveys were conducted with the project leader. Then the products were described and the quality characteristics and subcharacteristics required in the software product were identified. For the two software in our study, it was selected functionality as the required characteristic.
b. Next the product documents that would be subject to evaluation were identified.

c. The product users, analysts/developers and project leaders to be interviewed were identified. It was selected one user because the evaluated products represent a critical tool in his everyday work and at the same time this user has a good knowledge on the product requirements. There was only one project leader for both projects. It was identified one analyst/developer who works with both products.

d. The questionnaires were applied by two evaluators to the same group of users, analysts/developers and project leaders. Each evaluator was responsible for applying the evaluation to two specified software products.

6.3 Application of the Algorithm to obtain the Quality Estimate

The algorithm used to obtain the estimate of the quality of the software product is as follows:

1. Define the quality requirements depending on the interested party's evaluation needs. When dealing with evaluation of quality one must consider the circumstances. Each software product will have its own unique evaluation; therefore, assessing a product efficiency and effectiveness is only correct for one situation alone in one organization alone.
2. Identify what the party interested in the evaluation sees as the critical characteristics of the software product
3. Identify the critical subcharacteristics of the characteristics to be evaluated, associated with the “product effectiveness” dimension.
4. Apply the questionnaires associated with the critical subcharacteristics.
5. Check whether the evaluations made by the two evaluators have a high level of correlation. If not, see if the questionnaire contains any ambiguous question and if so ask those ones again, redefining the question.
6. Standardize the results of the questionnaires according to Table 11 (Rojas et al., 2000). There are two metrics type Rate and type N that must suffer a transformation from the original values to a Likert scale. Metrics with a Coverage Rate type are numbers that represent the satisfaction level regarding the requirements. It is a number that varies from 0 (it does not satisfy the requirement) and 1 (it satisfies the requirement completely). For example, the metric that deals with the Support

documentation for different users (programmers, developers, analysts, administrators, installers, etc.) is the result of dividing Number of existing support documents regarding users by number of support documents required for different users. Metrics type N represents a percentage metric regarding the minimum and maximum values. For example, when dealing with documented code, a metric can be the module percentage with comments in the heading.

Table 11. Standardization of the values obtained from the questionnaires according to type of metric

Type of metric	Value	Value Normalized
Likert	1	1
	2	2
	3	3
	4	4
	5	5
Rate	0 = < N < 0.25	1
	0.25 = < N < 0.50	2
	0.50 = < N < 0.75	3
	0.75 = < N < 1	4
	N =1	5
Flag	0	1
	1	5
N (Percentage)	0 = < N < 25	1
	25 = < N < 50	2
	50 = < N < 75	3
	75 = < N < 90	4
	90 = < N <=100	5

7. Check that 75% of the metrics fall within the optimum values (equal to or greater than 3) for each of the critical subcharacteristics. If not, a report is issued indicating that the product does not meet the basic requirements, and listing the main problems found. If 75% of the metrics have an optimum value, continue to the next step. This 75 % value was determined by the interested party in the evaluation so it can be modified depending on the requirements demanded of the product.
8. Calculate the rate of coverage of the critical subcharacteristics.
9. Obtain the ranking for each subcharacteristic. The values obtained in each subcharacteristic are calculated with the median.
10. Analyze by dimension and analyze how the product behaves in each dimension.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

6.4 Analysis of the results obtained

Once final data is obtained from the evaluation, these are analyzed to obtain results that will lead to relevant conclusions on the model's behavior. With this analysis it was possible to determine how the model estimates the quality of a given product, and at the same time to refine it.

The data were analyzed taking into account a) the characteristics and subcharacteristics of the two software products selected as priorities by the interested party in the evaluation, b) the comparison between the two software products, and lastly, c) an analysis of the behavior of the quality dimensions, for each product.

The model was validated using the stability measurement (reliability by test-retest) (Hernández et al., 1988). This measurement is used to assess the consistency of a subject's responses to specific scale items over time. Inconsistencies must be distinguished from changes in responses. To establish test retest reliability, an instrument is administered on two separate occasions, then the results are correlated. High positive correlation indicates good reliability**.**

The data obtained from these evaluations were analyzed using the coefficient of correlation. A positive correlation was obtained in the two products (Table 12), which indicates that the Evaluator factor did not lead to different evaluations for the results of the product evaluation.

Having ascertained the validity of the data obtained by the evaluation, each of the products was analyzed. As there are two evaluations for each product, it was necessary to produce a new set of metrics taking the greater value of both evaluations for each metric. This new set of metrics was used in the analysis of the results.

Table 12. Validation of the instrument with the stability measurement

Product	Stability measurement
P1	0.70
P2	0.68

7. Results

The results of the evaluation of the two software products were compared for this case study. Two comparisons were made. First of all the quality requirements specified by the client were considered. Next, the products were compared, taking into account the medians of the metrics by subcharacteristic.

7.1 Comparison of products P1 and P2 considering the quality requirements established

In this case, the quality requirements are established on the basis of the Functionality characteristic. The party interested in the evaluation indicated that a subcharacteristic is of an optimum level when 75 % of the associated metrics are equal to or greater than three.

It was shown that product P1 meets the quality requirements of Functionality, both in terms of product efficiency and effectiveness (see Table 13). By contrast, product P2 does not have an optimum level in the subcharacteristics of Interoperability and Security.

Table 13. Comparison of Products P1 and P2 as regards the Quality requirements of the Functionality characteristic

Dimension	Subcharacteristic	Product P1	Product P2
Product Effectiveness	Suitability	87.50	87.50
	Accuracy	90.00	100.00
	Interoperability	100.00	12.50
	Security	100.00	50.00
Product Efficiency	Correctness	100.00	100.00
	Structured	100.00	100.00
	Encapsulated	0.00	0.00
	Specified	100.00	100.00
Efficiency and Effectiveness of the Process	Processes related with Functionality	75.00	75.00
	Audit	50.00	50.00
	Quality Management	33.33	33.33

Since both products have the same systems development environment, there was 75% compliance by the processes related to product P1 and P2 functionality. However, lower levels than those expected in the audit and quality management were reached (see Figure 3).

To conclude, when it is considered necessary for 75 % of the metrics to have values equal to or greater than 3, both products have deficiencies as far as the "Encapsulated" subcharacteristic is concerned. Product P2 has lower levels of security than product P1. As far as the Interoperability of product P1 is concerned, it has a considerable advantage over product P2.

An important contribution by the model is that due to the establishment of requirements on the values of metrics (greater than or equal to 3) lower performance levels by products can be identified that would otherwise have been very hard to detect.

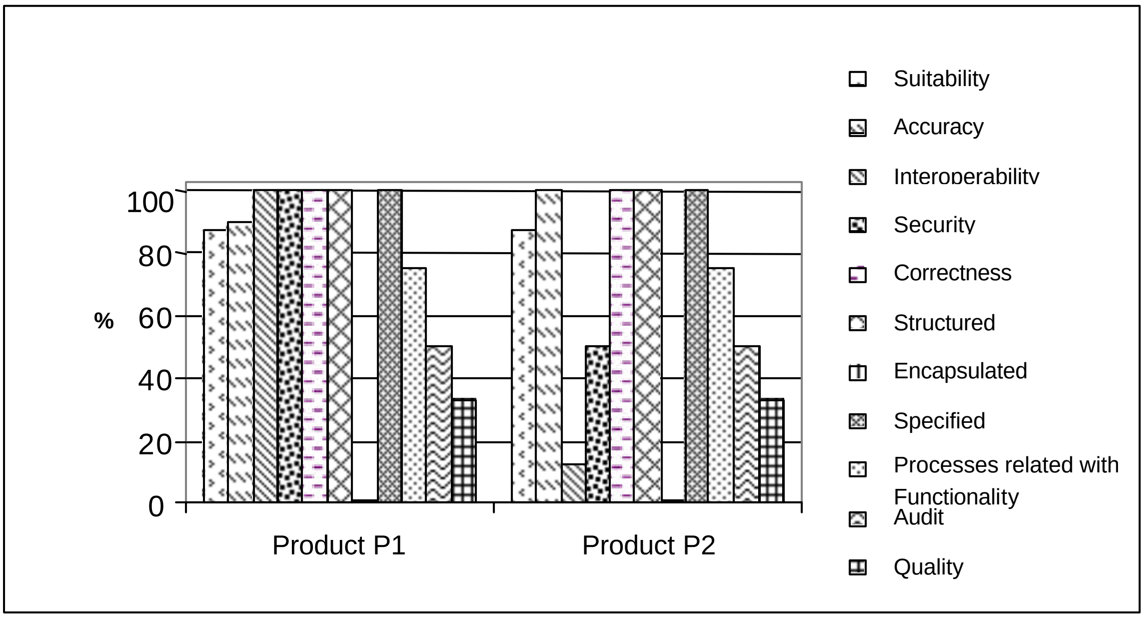

Figure 3. Comparison between Products P1 and P2 as regards the Quality requirements of the Functionality's subcharacteristics.

7.2 Comparison between Products P1 and P2 Medians of the Functionality subcharacteristics

Another way of presenting the results of the evaluation is to consider the medians by subcharacteristic, stressing that the values of the metrics are found within a range of 1 to 5 (Likert scale). The median is a measure of central tendency, tell us something about where the "middle" of the set is likely to be. The median is the middle-ranked item in the data set. That is, the median is the value *m* for which half the values in the data set are larger than *m* and half are smaller than *m* (Fenton and Pfleeger 1997). The ordinal scale used in our model brings information about an ordering of the classes or categories. The empirical relation system consists of classes that are ordered with respect to the attribute. The numbers represent ranking only, so addition, subtraction, and other arithmetic operation have no meaning (Fenton and Pfleeger 1997). Ordinal data has order, but the intervals between scale points may be uneven. Because of lack of equal distances, arithmetic operations are impossible with ordinal data, which are restricted to logical operations (more than, less than, equal than).

In this section, unlike the previous one, no minimum values in the values of the metrics are established. Table 14 and Figure 4 show more detailed values in relation to product behavior in each of the subcharacteristics. In the previous comparison in Figure 3, fulfillment of requirements in seven of the subcharacteristics was identical, but in the case of the comparison of the medians of all the metrics, only Accuracy kept the same ranking.

Using medians allowed to detect differences in processes in Products P1 and P2. We could identify a lower ranking in P2 processes. Interoperability and Security medians of Product P2 confirm the lower values shown in Figure 3 as regarding quality requirements. In general, we could observe that median measure is useful to show different behaviors of product P1 in relation with another product P2.

Table 14. Comparison between Products P1 and P2 regarding the Medians of the Functionality subcharacteristics

Dimension	Sub Characteristics	Median P1	Median P2
Product Effectiveness	Suitability	4	3
	Accuracy	5	5
	Interoperability	5	1
	Security	5	3
Product Efficiency	Correctness	5	4
	Structured	5	3
	Encapsulated	2	1
	Specified	4	4
Efficiency and Effectiveness	Processes related with Functionality	5	4
of the Process	Audit	3	3
	Quality management	2	2

Comparison of the products in two different ways showed that the establishment of optimum levels resulted in a product evaluation that reflects the quality of the product from the point of view of the party interested in the evaluation.

Having observed how the model behaved in two different products, it can be concluded that it enables weaknesses and strengths to be detected, thereby providing valid information that can be used to improve the products.

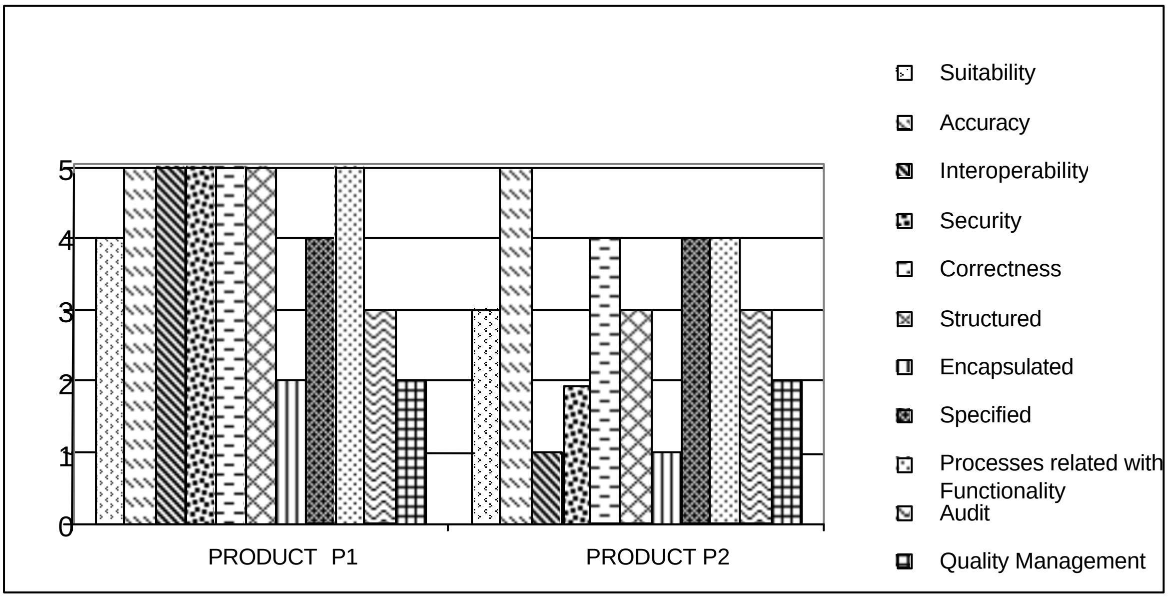

Figure 4. Comparison between Products P1 and P2 Medians of the Functionality's subcharacteristics.

8. Limitations of the empirical study

Every data collecting instrument must present two essential requirements: validity and reliability (Hernandez et al. 1998). Validity refers to the degree to which a study accurately reflects or assesses the specific concept that the researcher is attempting to measure. While reliability is concerned with the accuracy of the actual instrument or procedure, validity is concerned with the study's success at measuring what the researchers set out to measure. Considering reliability, we used inter-rater and test-retest reliability.

When talking about validity, it should be contemplated both external and internal validity. External validity refers to the extent to which the results of a study are generalizable or transferable. Components of external validity are: population of interest, situations of interest and societal/temporal changes. Population of interest refers to the following questions: Will the results generalize to others? In our study it was shown that when applying the model to other products they will show different quality levels adjusted to the case study requirements

Internal validity refers to (1) the rigor with which the study was conducted (e.g., the study's design, the care taken to conduct measurements, and decisions concerning what was and wasn't measured) and (2) the extent to which the designers of a study have taken into account alternative explanations for any casual relationships they explore in detail. In studies that do not explore causal relationships, only the first of these definitions should be considered when assessing internal validity (Writing at CSU 2003).

There are different types of internal validity: Construct Validity, Content Validity and others. Construct Validity seeks agreement between a theoretical concept and a specific measuring device or procedure. To understand whether a piece of research has construct validity, three steps should be followed. First, the theoretical relationships must be specified. Second, the empirical relationships between the measures of the concepts must be examined. Third, the empirical evidence must be interpreted in terms of how clarifies the construct validity of the particular measure being tested (Carmines and Zeller 1979). In this experiment, we did not consider construct validity, even though a recent research work (Pérez et al. 2001; Mendoza et al. 2003) dealt with this, looking for relationships between process and product quality.

Content Validity is based on the extent to which a measurement reflects the specific intended domain of content (Carmines and Zeller 1979). Content validity forces the researchers to define the very domains they are attempting to study. In our study it was relevant to look for the characteristics needed to obtain product quality. Different models were studied, and a complete set of characteristics was chosen to define the domains. With Dromey's Model the efficiency (internal quality) dimension was defined. Besides, there were identified some processes that support the product quality, considering the systemic quality.

One limitation in this model is that we can not compare two different characteristics in a particular product, for example functionality and efficiency of P1. But, it can be detected some problems in these characteristics individually.

Future research works must be oriented in adapting our model to new tendencies. Recently, research studies on software quality has been related to Component Based Development (Bertoa and Vallecillo 2002; Preiss and Wegmann 2002; OOSPICE 2002)

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

9. Conclusions

The Formulation of the Model enabled an applicable specification of the Systemic Quality Model to be attained - specifically of the product's Efficiency and Effectiveness dimension. It also provides an opportunity to study the processes that affect product quality.

The evaluation of two software products made it possible to ascertain that the model is an effective tool for analyzing product quality, that can be used to compare different software products and also to detect weaknesses in the product that need to be improved.

It revealed that Process Efficiency and Effectiveness influence product quality; an improvement in Auditing and Quality Management in particular has a repercussion on product improvements.

Applying questionnaires to users, analysts/developers and project leaders, enabled information to be obtained that could be used by them for product improvement.

The Systemic Quality model has a Process and a Product dimension, as in order for the quality evaluation to be systemic the Process dimension has to be incorporated. Further progress along these lines would be to incorporate the quality dimension into Personnel, considering the various characteristics it needs. This must be taken into account to complement this study.

Acknowledgement

The authors wish to thank the Risk Management team at Banco de Venezuela for providing the data which we used as case studies. This research received funding and an endorsement from CONICIT Project S1-2000000437, Universidad Simón Bolívar Project DI-CAI-S100094 and Universidad Ezequiel Zamora Project Nº 23101102.

References

Albrecht, A. and Gaffney, J. 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *Transactions on Software Engineering*, Vol. SE-9, N° 6, November: 639-648.

Bertoa, M. and Vallecillo,A. 2002. *Atributos de Calidad para Componentes COTS*. In Proc. of IDEAS'02, La Habana, Cuba: 352-363.

Boehm, B. W., Brown, J. R., Kaspar, H., Lipow M., McCleod, G. J. and Merritt M. J. 1978. *Characteristics of Software Quality*, Amsterdam, North Holland.

Callaos, N. and Callaos, B. 1996. Designing with a Systemic Total Quality. *Proceedings of the International Conference on Information Systems Analysis and Synthesis*, ISAS'96. Orlando,USA, pp. 15-23.

Carmines, E. and Zeller, R. 1979. *Reliability and validity assessment*, Newbury Park, Sage Publications.

Dromey, G. 1996. Cornering the Chimera, *IEEE Software*, January: 33-43.

Fenton, N. and Pfleeger, S. L. 1997. *Software Metrics: A Rigorous & Practical Approach*, Second edition, PWS Publishing Company.

Google. 2003. *Google Web Directory: Testing Services*, http://directory.google.com/Top/Computers/Software/Testing_Services/

Grady, R. and Caswell, D. 1987. *Software Metrics: Establishing a Company-Wide Program*, Prentice Hall.

Hernández , R.; Fernández, C. and Baptista , P. 1998. *Metodología de la Investigación*, Segunda Edición, McGraw-Hill.

Humphrey, W. S. 1989. *Managing the Software Process*. Addison-Wesley.

ISO/IEC 9126. 1991. *Information technology – Software product evaluation: Quality Characteristics and Guidelines for their use*, Geneva, ISO.

ISO/IEC 9126-1.2. 1998. *ISO/IEC 9126-1.2: Information Technology - Software Product Quality - Part 1: Quality Model*, ISO/IEC JTC1/SC7/WG6.

ISO/IEC TR 15504-2. 1998. *ISO/IEC TR 15504-2: 1998 (E) Information technology - Software process assessment -Part 2: A reference model for processes and process capability*. ISO/IEC JTC 1/SC 7. Canada.

McCall, J. A., Richards, P. K. and Walters G. F.1977. *Factors in Software Quality*, Vols I, II, III, AD/A-049-014/ 015/055. Springfield, VA: National Technical Information Service.

Mendoza, L. E., Pérez, M. A., Griman, A. C. and Ortega, M. 2003. *Análisis del Impacto del Proceso de Desarrollo en las Características de Calidad de Software*, IDEAS'2003, Asunción, Paraguay.

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

OOSPICE. 2002. http://www.oospice.com

Pérez, M. A., Rojas, T., Mendoza, L., and Grimán, A. 2001. *Systemic Quality Model for System Development Process: Case Study*, in Seventh Americas Conference on Information Systems – AMCIS 2001, D. Strong and D. Straub (eds.), Association for Information Systems, Boston: 1297-1304, http://www.lisi.usb.ve/publicaciones/calidad11.zip

Pfleeger, S. L. 1998. *Software Engineering: Theory and Practice*, Prentice Hall.

Preiss, O. and Wegmann, A. 2002. *A Systems Perspective on the Quality Description of Software Components*. 6th World Multiconference on Systemics, Cybernetics and Informatics. The 8th International Conference on Information System Analysis and Synthesis. Orlando.

Pressman, R. 1998. *Ingeniería de Software: Un enfoque Práctico*, McGraw Hill.

Rojas, T. and Pérez, M. 1995. A comparison of three Information System Development Methodologies Related to Effectiveness/Efficiency Criteria. *International Symposium on Applied Corporate Computing (ISACC'95)*, Monterrey, México.

Rojas, T; Pérez, M. A; Griman, A. C.; Ortega, M. and Díaz, A. M. 2000. *Modelo de decisión para soportar la selección de herramientas CASE*. Revista de la Facultad de Ingeniería, UCV, Vol. 15, Nº 2: 117-144.

Seijas, F. 1999. *Investigación por Muestreo*. Caracas, Ediciones Faces/UCV.

Voas, J. 1999. *Certification: Reducing the Hidden Costs of Poor Quality*, *IEEE Software*, July/August: 22-25.

Writing at CSU. 2003. *Overview: Reliability and Validity*. Writing Center at Colorado State University. http://writing.colostate.edu/references/research/relval/index.cfm

Maryoly Ortega. Assistant Professor at Ezequiel Zamora University. Phd student in the Department of Languages and Computer Science at Málaga University (Spain); MSc in Systems Engineering from Simón Bolívar University (2001); BSc degree in Computer Science from Central de Venezuela University (1992). Publications on: SCI'00, USA; SCI'02, USA; IDEAS'03, Paraguay. Revista de la Facultad de Ingeniería de la UCV 15(2). Research interests: Software Quality, Metrics, Software Engineering, CASE Tools, Component Based Software Engineering. Expertise areas: Information Systems

Reference: Ortega, Maryoly; Pérez, María and Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal*, 11:3, July 2003, pp. 219-242. Kluwer Academic Publishers, 2003

María A. Pérez de Ovalles. Member of the Association of Information Systems. Titular Professor at Simón Bolívar University. Ph.D. in Computer Science from Central de Venezuela University (1999); MSc. in Information Systems from Católica Andrés Bello University (1993); Electrical Engineering from Metropolitana University (1975). Publications on: ISACC'95, Mexico; AIS'96, USA; AIS'97, USA; AIS'98, USA; AMCIS '99, USA; CLEI '99, Uruguay; JOOP 12(6); AMCIS '00, USA; Journal Information & Software Technology, 2000; JOOP, 2000; SCI '00, USA; Revista de la Facultad de Ingeniería de la UCV 15(2); AMCIS '01, USA; JIISIC '01, Argentina; JIISIC ´02 , Brasil; AMCIS ´02 USA, SCI'02 USA, CLEI Electronic Journal ´02; IDEAS'03, Paraguay. Research interest: Process improvement, Software Engineering, Methodologies, CASE Tools, Information Technology. Expertise areas: Information Systems, Methodologies and Software Engineering.

Teresita Rojas. Titular Professor at Simón Bolívar University. MSc. in Business Administration from Simón Bolívar University (1986); Computer Engineering from Simón Bolívar University (1980), PPI: Level 1. Publications on: ISAS'95, Germany; ISACC'95, Mexico; AIS'96, USA; AIS'97, USA; IEEE'97 Man, Systems and Cybernetics, USA; AIS'98, USA; AMCIS '99, USA; AMCIS '00, USA; SCI '00, USA; Revista de la Facultad de Ingeniería de la UCV 15(2); AMCIS '01, USA; JIISIC '01, Argentina, Revista Colombiana de Computación ´01, JIISIC ´02 , Brasil, AMCIS ´02 USA, SCI'02 USA, CLEI Electronic Journal ´02, Software Quality Professional 02, Information Systems Management, 2003. Research interest: Improvement of the development process of Information Systems. Expertise areas: Information Systems development and organizational aspects, Methodologies, Quality, Case Studies in Venezuela.