

A Systemic Quality Model for Evaluating Software Products

Maryoly ORTEGA

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

María A. PÉREZ; Teresita ROJAS

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

ABSTRACT

Quality has become essential for ensuring that a company's products or processes meet customers' needs. Most product quality models focus on product characteristics. This article describes the design of a Quality Model with a Systemic Approach¹ to Software Products that assesses a product's efficiency and effectiveness considering processes implications on product characteristics. The 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.

1. INTRODUCTION

Certifications, standards and measurements must take into account a balanced proportion of processes, products and resource requirements [1, 2], 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 [3].

Given the complexity that dealing in detail with all the aspects involved in quality would involve, such as process, product, resources and personnel, the purpose of this article is to focus on the study of the product.

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 [3]. 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 [3].

The literature on the subject covers several product quality models, the best known in chronological order of appearance are: Mc Call (1977), Boehm (1978), FURPS (1987), ISO 9126 (1991) and 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 objective of this study is to identify the 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: the second section gives the background, the third describes the method used, the fourth the formulation of the model, the fifth the design of the survey with the case study, the sixth shows how the model is applied and the seventh the results analysis. The conclusions and recommendations are presented in the eighth and last section.

2. BACKGROUND

The literature on the subject contains a significant number of quality models that have gradually evolved: McCall Model, Boehm Model, FURPS, ISO 9126, Dromey Model, and Systemic Quality Model. By analyzing these quality models, it was possible to identify the advantages and disadvantages of these models when considering Software Quality Model with a Systemic Approach.

One of the major contributions of the Mc Call 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 [4, 5].

The Boehm model is similar to the Mc Call model in that it represents a hierarchical structure of characteristics, each of which contributes to total quality. Boehm's notion includes users needs, as Mc Call's does; however, it also adds the hardware yield characteristics not encountered in the Mc Call model [1].

One disadvantage of the FURPS model is that it fails to take account of the software product's portability [6].

One of the advantages of the ISO 9126 model is that it identifies the internal characteristics and external quality

¹ This research received funding and an endorsement from CONICIT Project S1-2000000437, Universidad Simón Bolívar Project DI-CAI-S100094 and Universidad Ezequiel Zamora Project N° 23101102.

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.

Dromey's 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.

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.

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 [7] and Voas [8]. Dromey says there no quality process can exist if it is not based on a product quality model.

A model that differs slightly from those previously studied, since it has no hierarchical structure, is the Callaos model [9]. 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.

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.

In order to build a product quality model with a systemic approach, different dimensions must be considered, involving a high degree of complexity.

- 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, code).
- The quality model prototype must be applied by two evaluators to two software products, so it can be refined.

The case study is 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.
- 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.

4. FORMULATION OF THE MODEL

This section presents the formulation of the Software Product Quality Model prototype based on the Systemic Quality [9]. 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 Mc Call 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 [10], specifically as regards the external characteristics defined, which take into account how the product behaves in its environment: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability.

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 for evaluating software development processes. However, only the processes covered by ISO 15504 [11], which favor software quality characteristics will be considered within the framework of this research. The resulting model is shown in Figure N° 1.

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.

Reliability is the capacity of a software product to maintain a specified level of performance when used under specific conditions.

Usability is the capacity of a software product to be attractive, understood, learned and used by the user under certain specific conditions.

Efficiency is the capacity of the software product to perform adequately under specific conditions, depending on the amount of resources used.

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.

Portability is the capacity of the software product to be transferred from one environment to another.

Having defined the subcharacteristics associated with each quality dimension [12], 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 characteristic is obtained aggregating the subcharacteristics values. 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 N° 1 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.

Table N° 1. 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= Monthly
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 [10]

** Adaptation [11]

It is important to point out here that there are two processes, Auditing and Quality Management, that affect each of the quality characteristics speci-4,d within the model. For example, the Auditing process has the following metrics [11]: 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?

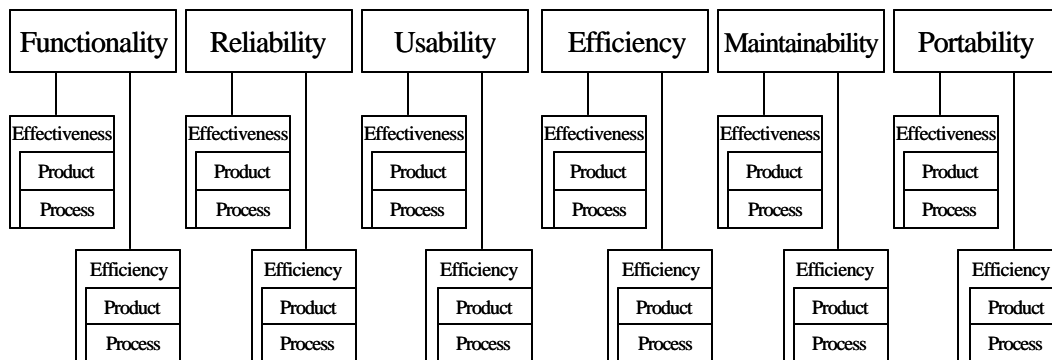


Figure N° 1. Elements of the Systemic Quality model for Software Products

5. 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".

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:

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.

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

Previous to the application of the algorithm to obtain the quality estimate two steps were followed: a) Identification, Preparation and Validation of the Measurement Instrument and b) Application of the Instrument to two Software Products and Refinement of the Model

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.
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.
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. Calculate the final ranking, taking into account all the metrics values associated with the essential subcharacteristics. To obtain the ranking for each characteristic, the values obtained in each subcharacteristic are added up.
10. Analyze by dimension and analyze how the product behaves in each dimension.

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) [13]. The data obtained from these evaluations were analyzed using the coefficient of correlation. A positive correlation was obtained in the two products (Table N° 2), which indicates that the Evaluator factor did not lead to different evaluations for the results of the product evaluation.

Table N° 2. Validation of the model with the stability measurement

Product	Stability measurement
P1	0.70
P2	0.68

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, each of the metrics obtained by the two evaluators had to be averaged. This produced a new set of metrics resulting from averaging the two evaluations. This new set of metrics was used in the analysis of the results.

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. Then the products were compared, taking into account the sum of the metrics in their entirety, regardless of whether they exceeded the values required by the party interested in the evaluation.

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. By contrast, product P2 does not have an optimum level in the subcharacteristics of Interoperability and Security. 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 2).

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.

Comparison between Products P1 and P2 Grand Total of the Functionality characteristic

Another way of presenting the results of the evaluation is to consider the sum of the values obtained in each of the metrics (Grand Total), stressing that the values of the metrics are found within a range of 1 to 5. Therefore, 100 % would be made up of a value of 5 in all the metrics. In this section, unlike the previous one, no minimum values in the values of the metrics are established.

Figure 3 shows more detailed values in relation to product behavior in each of the subcharacteristics. In the previous comparison in Figure 2, fulfillment of requirements in seven of the subcharacteristics was identical, but in the case of the comparison of the Grand Total of all the metrics, differences in these subcharacteristics can be seen. Suitability, correctness and structuring, in particular, are higher than in product P1. The case of the Accuracy subcharacteristic is special in that product P2 meets the highest number of requirements in this subcharacteristic (see Figure 2). However, if all the values of the metrics are taken into account, product P1 gets a higher score, which means that the Accuracy subcharacteristic in general had better values in product P1, except that product P2 exceeded the optimum values in a larger number of metrics.

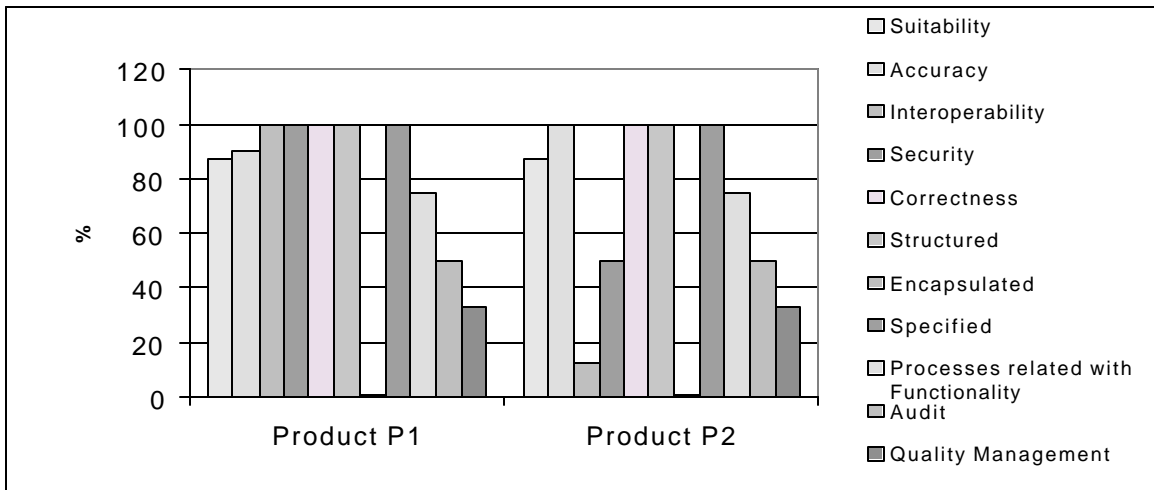


Figure N°2 . Comparison between Products P1 and P2 as regards the Quality requirements of the Functionality characteristic

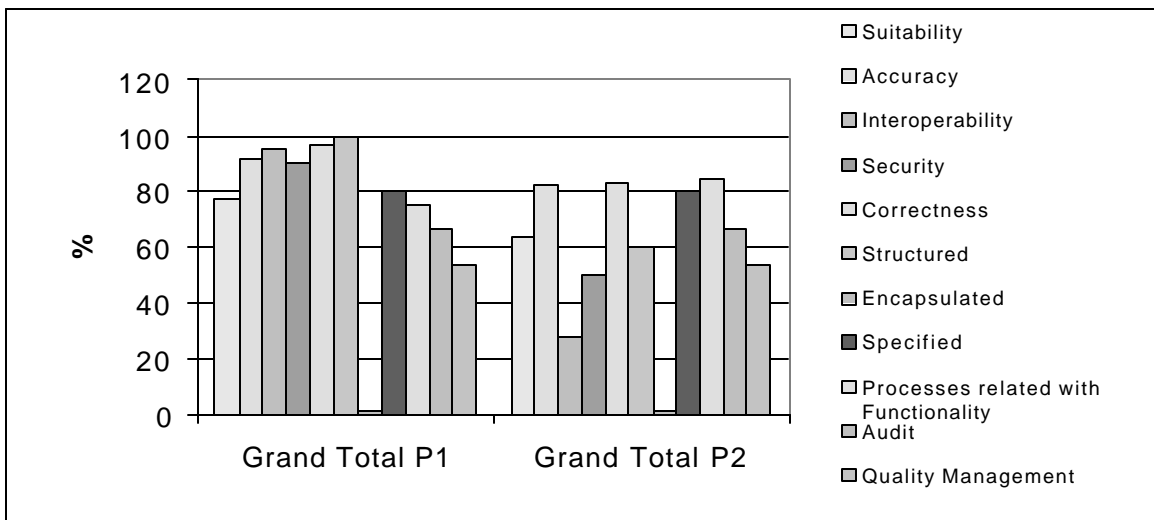


Figure N°3 . Comparison between Products P1 and P2 Grand Total of the Functionality characteristic

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 informally, thereby providing valid information that can be used to improve the products.

8. CONCLUSIONS

Compared with existing models the Formulation of the Model is an original contribution, since this model enables 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.

One of the most important aspects of the model was that it separated each of the software product quality characteristics proposed by ISO 9126 into Product Effectiveness and Efficiency dimensions, and identified the processes that influence these characteristics.

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. This information is not present in previous models, what represents a contribution in this area.

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.

9. ACKNOWLEDGEMENT

The authors wish to thank the company used as a case study for its invaluable help. This research received funding and an endorsement from CONICIT Project S1-2000000437, Universidad Simón Bolívar Project DI-CAI-S100094 and Universidad Ezequiel Zamora Project N° 23101102.

10. REFERENCES

- [1] S. L. Pfleeger, *Software Engineering: Theory and Practice*, Prentice Hall, 1998.
- [2] W. S. Humphrey, *Managing the Software Process*, Addison-Wesley, 1989.
- [3] N. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous & Practical Approach*, PWS Publishing Company, Second edition, 1997.
- [4] A. C. Gillies, *Software Quality: Theory and Management*, Thomson Computer Press, Second edition, 1997.
- [5] R. Pressman, *Ingeniería de Software: Un enfoque Práctico*, McGraw Hill, 1998.
- [6] R. Grady and D. Caswell, *Software Metrics: Establishing a Company-Wide Program*, Prentice Hall, 1987.
- [7] G. Dromey, "Cornering the Chimera", *IEEE Software*, January, 1996, pp. 33-43.
- [8] J. Voas, "Certification: Reducing the Hidden Costs of Poor Quality", *IEEE Software*, July/August, 1999, pp. 22-25.
- [9] N. Callaos and B. Callaos, "Designing with a Systemic Total Quality", *Proceedings of the International Conference on Information Systems Analysis and Synthesis*, ISAS'96, Orlando, USA, 1996, pp. 15-23.
- [10] ISO/IEC 9126-1.2: *Information Technology - Software Product Quality - Part 1: Quality Model*, ISO/IEC JTC1/SC7/WG6, Junio, 1998.
- [11] ISO/IEC TR 15504-2: (E) *Information technology - Software process assessment-Part 2: A reference model for processes and process capability*, ISO/IEC JTC 1/SC 7, Canada, 1998.
- [12] M. Ortega; M. A. Pérez and T. Rojas, "A Model for Software Quality with a Systemic Focus", *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Volume II, Orlando, USA, 2000, pp.464-469.
- [13] R. Hernández; C. Fernández and P. Baptista, *Metodología de la Investigación. Segunda Edición*, Mac Graw Hill, 1998.