Construction of a Quality Model for Software Development based on the Family of ISO / IEC 25000 Standards for Generating Web Applications

Ing. Carolina Tlapalamatl Lopez[1], M.C. Juan Ramos Ramos[2], Dr. José Juan Hernández Mora[3], Ing. Is-boset Alvarado[4]

[1, 2, 3, 4] Tecnológico Nacional de México / Instituto Tecnológico de Apizaco, Carretera Apizaco Tzompantepec, esquina con Av. Instituto Tecnológico S/N, Conurbado Apizaco - Tzompantepec, Tlaxcala, Mex. C.P.90300

Abstract: ***This work shows the proposal of a Quality Model to develop software taking as reference the guidelines established by the family of ISO / IEC 25000 Standards, seeking to simplify the evaluation process, developing a series of activities and procedures aimed at achieving the product characteristics, satisfying the corresponding requirements and metrics. The conjugation of the proposed model with a set of tools that facilitate the referred process is also presented, achieving good practices when developing software, resulting in quality products with adequate elements of security and maintainability. This model seeks to define a reference framework to define and evaluate the required metrics as software quality factors, incorporating the various items included within the standards included in ISO / IEC 25000, such as the specific characteristics of a quality model. , the specification of requirements, the definition of benchmarks for quality measurement and evaluation, as well as additional common models, terms and definitions.***

Keywords: Rules ISO/IEC 25000, Software quality, Quality model for software development.

1. Introduction

Software development is one of the activities in greatest demand in recent decades. Every day technology evolves and Software Engineering requires constant updating incorporating quality standards and metrics, including the implementation of models that serve as a guide and the use of tools that help improve testing, maintenance, usability, portability and compatibility, among other required characteristics.

Currently, several quality norms and standards are known that have facilitated efficiency, delivery and good practices, however, it is vitally important to have a quality model that guides processes to have better results, as well as the use of tools. That help the evaluation process in the different stages involved in software development.

The family of ISO / IEC 25000 Standards emerges as an evolution of other standards such as ISO / IEC 9126 and ISO/ IEC 14598. Its task is to define a common framework for evaluating the quality of software products. It includes the peculiarities of a quality model, specification of requirements, references for quality measurement and evaluation, as well as common models, terms and definitions [1].

The ISO / IEC 25000 standards therefore have the objective of creating a working environment to define the requirements and evaluate the quality of the software product, replacing the previous ISO / IEC 9126 and ISO / IEC 14598 and thus becoming the benchmark to follow.

2. Theoretical Elements

2.1 Software quality management

"Quality management is an inseparable philosophy of the customer's needs and the goals of the company when developing products that require quality assurance, guarantees maximum efficiency within the company and strengthens commercial leadership through the implementation of processes and systems that promote excellence, while preventing errors and ensuring that all the firm's goals are achieved without wasting effort. "This is the definition provided by the British Quality Association (BQA), in 1989, regarding the techniques involved in the Total paradigm Quality Management (TQM) [2]. This concept is fully applicable when developing software, since the client's requirements will always be the triggering element that guides developers to generate any computer application, reducing errors, taking maximum advantage of resources and seeking to achieve the established objectives.

2.2 Software Quality

The term can be ambiguous and even subjective because, like beauty, quality depends on who observes it. It is necessary to define the concept clearly, since, if the quality cannot be defined, it cannot be measured; and where the quality cannot be measured then it cannot be controlled.

Within the context of Software Engineering, one of the definitions of quality in software is that proposed by the International Standards Organization through the family of ISO / IEC 25000 Standards: “All the characteristics of a software product that have as an ability to meet explicit or implicit needs” [2].

Licensed Under Creative Commons Attribution CC BY

There are several definitions, however, they agree that the software has quality if it meets or exceeds the user's expectations regarding [3].

- Functionality (that serves a purpose),
- Execution (which is practical),
- Reliability (do what you should),
- Availability (that works under any circumstances)
- Support, at a cost less than or equal to what the user is willing to pay.

Software quality refers to "The factors of a software product that contribute to the complete and total satisfaction of the needs of a user or organization” [3].

Modelos a nivel de proceso

Based on the information gathered, a timeline is presented through *Figure 1* that incorporates some process-level models that support this document.

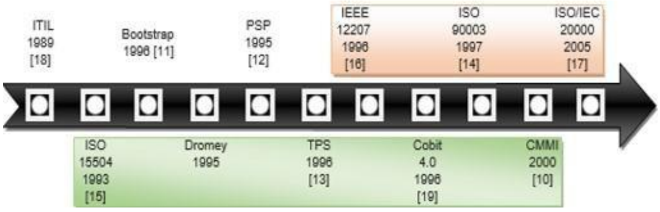

Figure 1: Process level model timeline (FOwn source)

The ISO / IEC 25000 family of standards and their dimensions.

Software quality is a very broad concept, which is made up of several dimensions. These dimensions according to the quality model described in ISO / IEC 25010 are as follows [6] [7]:

- **Functional Fitness**: The degree to which the product provides the functions that meet the implicit and explicit needs when the product is used under certain conditions.
- **Performance efficiency:** performance relative to the amount of resources used under certain conditions.
- **Compatibility:** condition that makes a program and a system, architecture or application to understand each other correctly, directly or indirectly (through an algorithm).
- **Usability:** effort required by the user to use the product satisfactorily.
- **Reliability:** the software's ability to maintain the required system performance, for a set time and under a set of defined conditions.
- **Security:** the software's ability to meet the risk levels allowed for both physical damage and possible data risks.
- **Maintainability:** effort required to adapt to new specifications and software requirements.
- **Portability:** ability of software to be transferred from one environment to another.

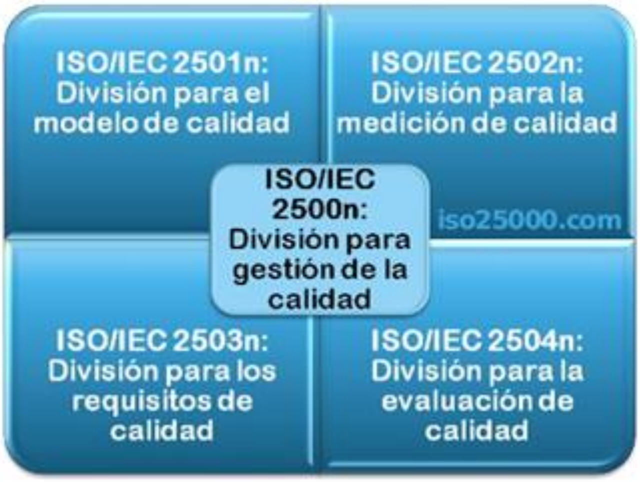
ISO/IEC 2501n: División para el modelo de calidad
ISO/IEC 2502n: División para la medición de calidad
ISO/IEC 2500n: División para gestión de la calidad
iso25000.com
ISO/IEC 2503n: División para los requisitos de calidad
ISO/IEC 2504n: División para la evaluación de calidad

Figure 2: Family Quality model ISO/IEC 2500 (own: ISO)

3. Sciption of the Proposed Model

Research and Impact

Economic losses and quality problems in software can commonly be derived from the lack of a reference methodology and metrics that streamline and provide success for the evaluation. On the other hand, and according to the latest report, The CHAOS stated of the Standish Group only 39% of the software development projects finished in the estimated time, with the planned resources and with an acceptable quality [5].

In this context, activities related to software quality are becoming increasingly important, mainly due to two factors: The first is the increase in outsourcing, which leads to client companies having to evaluate and control the quality of the software products delivered by the developer companies. They have to have the necessary resources to ensure that the product they develop will meet customer expectations.

The second is the proliferation of standards and certifications related to software quality, such as ISO / IEC 15504 and ISO/ IEC 12207 (SPICE) or models such as CMMI (Integrated Capacity and Maturity Model). However, the idea that evaluations of software should be based on direct evidence of the product itself and not only on evidence of the processes used for its construction has spread throughout the sector.

Prototyping of the rule family

The ISO / IEC 25000 is articulated in several divisions, including ISO / IEC 25040, which defines the process for evaluating the quality of the software product, and ISO / IEC 25010, which determines the quality characteristics and sub- characteristics that can be evaluate for a software product.

- The most important reasons why an organization may be interested in evaluating its product according to ISO / IEC 25000 may include the following:
- Differentiate yourself from competitors, ensuring delivery times and reducing product failures after implementation in production.
- Establish agreements in the field of service, defining certain quality parameters that the product must meet before being delivered.

Licensed Under Creative Commons Attribution CC BY

- Detect defects in the software product and proceed to eliminate them before delivery, which means cost savings in the subsequent maintenance phase.
- Evaluate and control the performance in the developed software product, ensuring that you can generate the results taking into account the time and resource restrictions established.
- Ensure that the software product developed respects the levels necessary for the security characteristics (confidentiality, integrity, authenticity, non-repudiation, etc.).
- Check that the developed product can be put into production without compromising the rest of the systems and maintaining compatibility with the necessary interfaces [8], [9], [10].

Through *Figure 3* it is possible to observe the prototyping of the family of ISO / IEC 25000 Standards, representing its wide range of stages to consider to have in order the processes during the framework, this will avoid failures during the planning process, evaluation, maintenance, portability, among others, that could generate delays and failures in the short or long term.

Plan development

The quality model depends on the commitment of all levels of the ISO / IEC 25000 Standard, especially the group in charge of managing quality and based on the requirements of the development of any application. This reference is the basis of the proposed model, seeking at all times the implementation of the management of a quality system based on the family of ISO / IEC 25000 Standards. The model allows to develop software projects analyzing the requirements provided by the client, defining, managing, controlling and continuously improving all processes. It also makes it possible to count on the participation of the entire organization, group or team, depending on the case.

To develop each of these stages, a set of objectives, tasks, managers and necessary tools are proposed, as shown in *Figure 4*, which describes each of the elements of the resulting model.

4. Conclusions

The family of ISO / IEC 25000 quality standards managed as a quality model is suitable for evaluating any system or web application.

The proposed model details the reference processes carried out for the evaluation of a software system with quality metrics required for its functionality, since security, usability, compatibility and maintenance metrics were required, among others.

The proper management of the different standards will establish the sequence for quality assurance, using various tools that help to have better results. It is successful if it is carried out seriously by the people responsible for leading the software quality assessment projects. It improves the maturity and performance of organizations, but does not compensate for poor management or wrong strategic decisions.

By implementing this quality model supported by the ISO / IEC 25000 family of standards, it can be seen that the complementary stages are a fundamental basis for managing an appropriate quality system, as well as the tasks that will lead to responsible compliance and execution.

The management of this model guarantees that the quality measurement tests will considerably reduce any type of error, avoiding losses, generating positive results and, in the event of errors, taking the necessary measures by means of the immediate resolution model that will be carried out avoiding than the collapse project, since each stage is measured as the software development progresses.

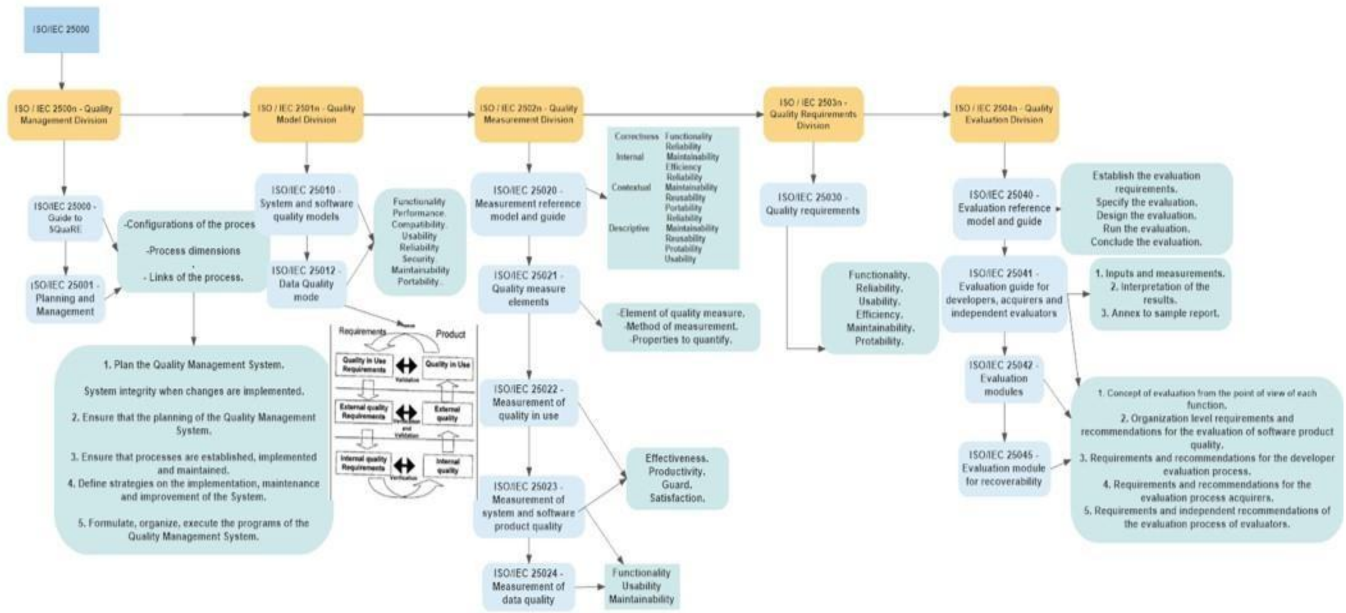
ISO/IEC 25000
ISO / IEC 2500n - Quality Management Division
ISO / IEC 2501n - Quality Model Division
ISO / IEC 2502n - Quality Measurement Division
ISO / IEC 2503n - Quality Requirements Division
ISO / IEC 2504n - Quality Evaluation Division
ISO/IEC 25000 - Guide to SQuaRE
ISO/IEC 25001 - Planning and Management
-Configurations of the proces
-Process dimensions
- Links of the process.
1. Plan the Quality Management System.
System integrity when changes are implemented.
2. Ensure that the planning of the Quality Management System.
3. Ensure that processes are established, implemented and maintained.
4. Define strategies on the implementation, maintenance and improvement of the System.
5. Formulate, organize, execute the programs of the Quality Management System.
ISO/IEC 25010 - System and software quality models
ISO/IEC 25012 - Data Quality mode
Functionality
Performance.
Compatibility.
Usability
Reliability
Security
Maintainability
Portability.
Requirements
Product
Quality in Use Requirements
Quality in Use
Validation
External quality Requirements
External quality
Verification and Validation
Internal quality Requirements
Internal quality
Verification
ISO/IEC 25020 - Measurement reference model and guide
Correctness
Internal
Contextual
Descriptive
Functionality
Reliability
Maintainability
Efficiency
Reliability
Maintainability
Reusability
Portability
Reliability
Maintainability
Reusability
Protability
Usability
ISO/IEC 25021 - Quality measure elements
-Element of quality measure.
-Method of measurement.
-Properties to quantify.
ISO/IEC 25022 - Measurement of quality in use
ISO/IEC 25023 - Measurement of system and software product quality
Effectiveness.
Productivity.
Guard.
Satisfaction.
ISO/IEC 25024 - Measurement of data quality
Functionality
Usability
Maintainability
ISO/IEC 25030 - Quality requirements
Functionality.
Reliability.
Usability.
Efficiency.
Maintainability.
Protability.
ISO/IEC 25040 - Evaluation reference model and guide
Establish the evaluation requirements.
Specify the evaluation.
Design the evaluation.
Run the evaluation.
Conclude the evaluation.
ISO/IEC 25041 - Evaluation guide for developers, acquirers and independent evaluators
1. Inputs and measurements.
2. Interpretation of the results.
3. Annex to sample report.
ISO/IEC 25042 - Evaluation modules
ISO/IEC 25045 - Evaluation module for recoverability
1. Concept of evaluation from the point of view of each function.
2. Organization level requirements and recommendations for the evaluation of software product quality.
3. Requirements and recommendations for the developer evaluation process.
4. Requirements and recommendations for the evaluation process acquirers.
5. Requirements and independent recommendations of the evaluation process of evaluators.

Figure 3: Prototyped Family ISO (Own source)

Licensed Under Creative Commons Attribution CC BY

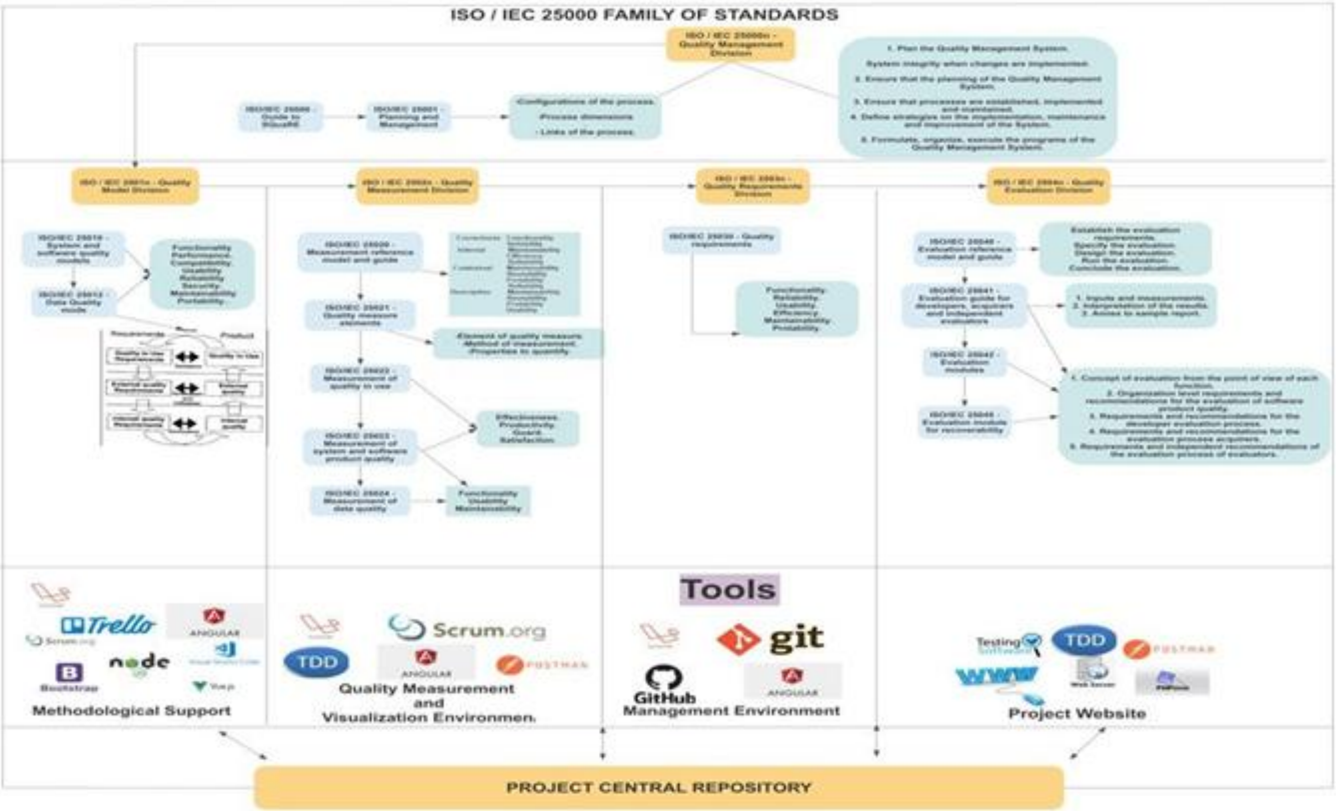
ISO / IEC 25000 FAMILY OF STANDARDS
ISO / IEC 25000n - Quality Management Division
ISO / IEC 2501n - Quality Model Division
ISO / IEC 2502n - Quality Measurement Division
ISO / IEC 2503n - Quality Requirements Division
ISO / IEC 2504n - Quality Evaluation Division
Tools
Trello
Scrum.org
ANGULAR
node
Methodological Support
Scrum.org
TDD
ANGULAR
POSTMAN
Quality Measurement and Visualization Environmen
git
GitHub
ANGULAR
Management Environment
Testing Software
TDD
POSTMAN
WWW
Web Server
Project Website
PROJECT CENTRAL REPOSITORY

Figure 4: Family Quality Model ISO/IEC 2500 (Own source)

References

[1] J. Alexander, «Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000 Integrating software development frameworks: Scrum, PSP and ISO25000.,» Ventana Informática, vol. 32, pp. 151-164, 2015.

[2] D. G. G. Roberto Carro Paz, «Administración de la calidad total,» 2010.

[3] B. Leydi, «"Modelos De Calidad De Software" Tesis: Control de Calidad de software,» Ica Peru, 2012.

[4] P. M. V. D. CASTELLANOS, «GEXRENOF: Herramienta para la gestión de pruebas no funcionales basada en el estándar ISO/IEC 25000, Universidad de las Ciencias Informáticas (UCI),» de 11 Congreso Multidisciplinario de ciencias aplicadas, Cuba, 2014.

[5] AENOR, «Calidad del producto software,» 2013.

[6] D. CIFUENTES, «Modelos de la calidad de software,» ranscripción de Modelos de la calidad de software, 2015.

[7] J. GARZÁS, «Cómo realizamos nuestra primera implantación real de la ISO 25000,» año 2007. [En línea].Available: http://www.javiergarzas.com/2014/04/iso-25000-implantacion.html.

[8] E. Balseca, «Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000 (Trabajo de fin de grado),» Escuela Politécnica Nacional, 2014. [En línea]. Available:

[9] http://bibdigital.epn.edu.ec/handle/15000/9113.

[10] J. M. R. P. A. E. S. B. M. &. P. Calabrese, «Assistant for the Evaluation of Software Product Quality Characteristics Proposed by ISO/IEC 25010 Based on GQM-Dened metrics>> Argentine Congress of Computer Science, Springer, Argentina, 2018.

[11] E. y. a. d. u. m. d. c. e. u. d. p. w. d. l. b. d. t. d. l. P. U.

[12] C. d. P. b. e. l. n. I. 2. y. f. (. d. fi. d. grado),

[13] «Universidad Católica del Perú basado en la norma ISO/IEC 25000 y familia (Trabajo de fin de grado),»

[14] ontificia Universidad Católica del Perú, Lima, Peru, 2005.

Author Profile

Carolina Tlapalamatl López she has a degree in Engineering in Information and Communication Technologies from the Tecnológico Nacional de México (TecNM) / Instituto Tecnológico de Apizaco, since 2018. She is currently studying a master's degree in Computer Systems in Software Engineering from the TecNM / Instituto Tecnológico de Apizaco.

Juan Ramos Ramos has a degree in Computer Science from the Instituto Tecnológico de Apizaco, from 1993. He is also a Master in Computer Science and Telecommunications from the Instituto de Estudios Universitarios, A.C.; he works as a full-time professor at the Tecnológico Nacional de México (TecNM) / Instituto Tecnológico de Apizaco in the area of Systems and Computing, teaching at the undergraduate and postgraduate level, in the areas of Programming and Software Engineering.

José Juan Hernández Mora has a Doctor of Teaching Excellence degree from the University of Los Angeles, 2019. He also has a degree in Computer engineer from the Universidad Autónoma de Tlaxcala, from 1994. Master in Computer science at the National Center for Research and Technological Development of the TecNM, 2003. Research professor at the Instituto Tecnológico de Apizaco del TecNM. Teacher of the Master of computer systems of the TevcNM / Instituto Tecnológico de Apizaco.

Licensed Under Creative Commons Attribution CC BY

Is-boset Jiménez Alvarado has a degree in Engineering in Information and Communication Technologies from the Instituto Tecnológico de Apizaco de Tlaxcala, from 2016. He is currently studying the masters in Computer Systems in Software Engineering from the TecNM / Instituto Tecnológico de Apizaco.

Licensed Under Creative Commons Attribution CC BY