Software Quality Assessment Applied for the Governmental Organizations using ISO/IEC 25000

Roberto Carlos Bautista Ramos
Centro de Posgrados
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
Departamento de Ciencias de la Computación
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
rcbautista1@espe.edu.ec

Sang Guun Yoo
Departamento de Informática y Ciencias de la Computación
Escuela Politécnica Nacional
Quito, Ecuador
Departamento de Ciencias de la Computación
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
sang.yoo@epn.edu.ec, yysang@espe.edu.ec

Nancy Velásquez Villagrán
Centro de Posgrados
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
nancy.velazquez87@hotmail.com

Geovanni Ninahualpa Quiña
Departamento de Ciencias de la Computación
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
gninahualpa@espe.edu.ec

***Abstract*—The objective of this paper is to share our experience in applying the ISO/IEC 25000 standard for the evaluation of software quality requirements. In this case study, Outlook Web Access (OWA) and Thunderbird will be evaluated as possible e-mail solutions for al real governmental institution, considering their technical and functional characteristics such as assistance for configuring mail accounts, address book, message file, robust privacy, protection against identity spoofing, automatic update, garbage removal, integration with Exchange and folder management to back up information. In this scenario, Outlook Web Access and Thunderbird have been selected as candidate's due to their main advantages e.g. integration with Microsoft Exchange and Open source characteristic, respectively. This work provides an interesting illustration of ISO/IEC 25000 implementation in a Governmental Institution detailing its benefits as a method to assessing software quality.**

Keywords—electronic mail; ISO/IEC 25000; software quality; software assessment.

I. Introduction

Currently, companies and institutions incorporate Information and Communications Technology (ICT) to solve the different needs of the organization, to comply with laws and regulations, and to satisfy the growing demand for transactions and operations among its internal users and external entities. The selection and evaluation of computer solutions can be done empirically, based on the knowledge of a specialist and advices of suppliers [1].

Fenton, N. and Pfleeger S, [2] indicates that there is a need for a methodological and technological environment to make automatic the process of acquisition, quality measurement, and analysis and the presentation of results of a software product. In other words, the quality of a software product is fundamental for the fulfillment of the functionality expected by the final user and for its adequate installation and administration.

With this background, the present work has the purpose of sharing an experience of selection and evaluation of computer solutions for a real public institution using an international standard [3]. This work applied a methodological process to comply with the administrative, technological and regulatory requirements of the organization, and it has considered the functional and technical characteristics of the candidate solutions.

The rest of the paper is organized as follows. Section 2 details some concepts used in this paper. Then, section 3 details the process of assessing the software quality of an organization. Finally, section 4 concludes this paper.

II. Background

A. SQUARE Common Model

In 2005, the International Standard ISO/IEC 25000, also known as SQuaRE (Systems and Software Quality Requirements and Evaluation), which replaces ISO/IEC 9126 (Software Quality Assessment) and ISO/IEC 14598 (Software Product Evaluation) was launched. The objective of creating SQuaRE was to cover two important processes i.e. the specification of requirements and the evaluation of software's quality, leveraged by a process of measuring software quality [3]. In addition to helping those software products in development or acquisition, it should be noted that the series of SQuaRE Standards are dedicated solely to emphasize the quality of the software product. [4 - 5].

The standard ISO/IEC 25000 establishes criteria for the specification of quality requirements of software products, their metrics and their evaluation, and includes a quality model to unify the definitions of customer quality with the attributes in the development process [3]. It is important to mention that ISO/IEC 25000 has been used for the delivery of new ideas for Software Evaluation such as [16-19]

978-1-5386-2521-7/18/$31.00 ©2018 IEEE

In case of Ecuador, ISO/IEC 25000 was adopted locally through the technical standard called NTE INEN-ISO/IEC 25000 Systems and Software Quality Requirements and Evaluation [3].

SQuaRE standard is comprised of the following models.

General Reference Model: The general reference model was created to help users navigate through the different SQuaRE standards (see Fig. 1). The choice of appropriate standards and documents of the SQuaRE series [6] depends on the user's role and information needs.

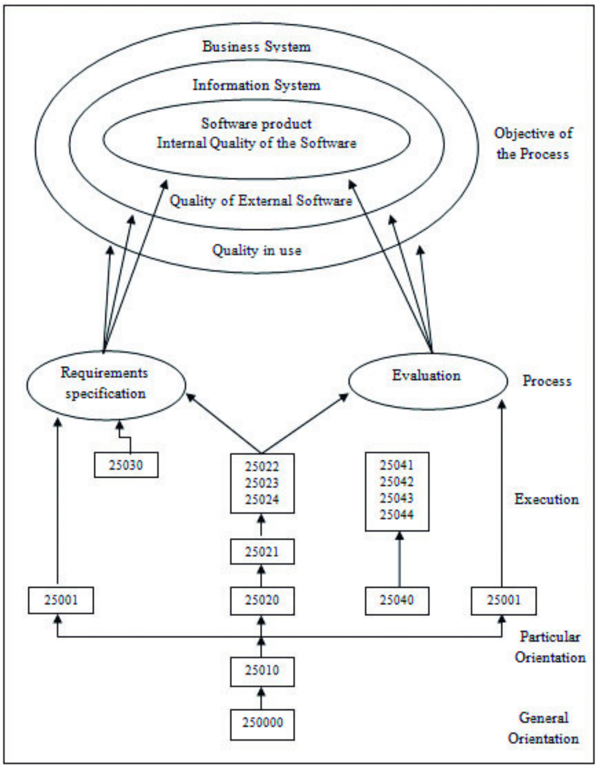

Fig. 1. SQUARE Model

Software Product Quality Life Cycle Model: According to the NTE ISO / IEC 25000 standard, the software product has a life cycle composed of three phases. The first phase focuses on the development of the product and represents the internal quality of the software; the second phase is associated to the product in operation and represents the external quality of the software; and the last phase corresponds to the product in use, which represents the quality of the software in use [6].

Quality Model Structure: The SQuaRE quality model classifies the software quality into characteristics which are divided into sub-characteristics and quality attributes (see Fig. 2). SQuaRE's quality model consists of two parts: the model for External and Internal Software Quality and the model for Quality in Use [7].

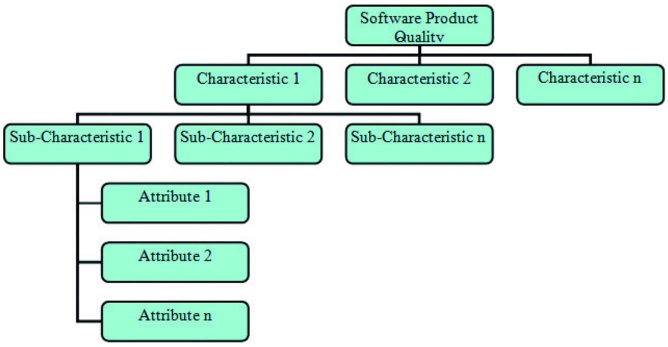

Fig. 2. Quality Model Structure

B. *Quality Model*

According to ISO 8402 (1986), a quality model can be defined as the set of quality factors and relationships between them, which provides a basis for the specification of quality requirements and for the evaluation of the quality of software components [7].

According to Dr. Witold Suryn, a professor at the School of Higher Technology in Montreal, Canada, quality models present an approach to unite different quality attributes with the basic objectives of: helping to understand how several facets of quality contribute to the whole and emphasize how software quality is much more than just defects and failures correction [15]. In addition, the models represent an aid to navigate the map of quality characteristics, sub-characteristics and appropriate measures, and they help to define the evaluation profile.

C. *Quality Model Framework*

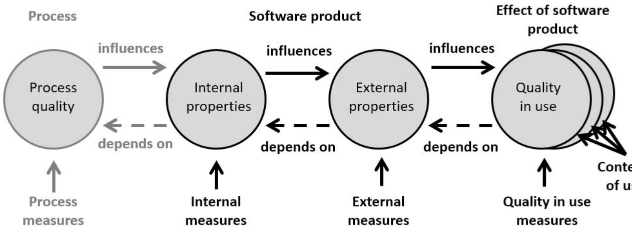

Fig. 3. Quality and the Life Cycle

According to NTE INEN-ISO/IEC 25000 [8], the Quality Model Framework explains the relation among different quality approaches (see Fig. 3).

The quality of the software product can be evaluated by measuring the internal, external and in use qualities of the software. The quality in use depends on the external behavior and the latter obeys the internal properties of the software.

D. *Software Quality Evaluation Process using ISO/IEC 25040* [9]*:*

The evaluator must perform the activities described below:

- To establish the evaluation requirements, which include the tasks of: (1) establishing the purpose of the evaluation, (2) obtaining the quality requirements of the software product, and (3) identifying the parts of the product to be included in the evaluation.

- To specify the evaluation, which includes the tasks of: (1) selecting quality measures (evaluation modules), (2) defining the decision criteria for quality measures, and (3) establishing the decision criteria for the evaluation.
- To design the evaluation, which includes the task of planning the evaluation activities.
- To execute the evaluation, which includes the tasks of: (1) making measurements, (2) applying the decision criteria for quality measures, and (3) applying the decision criteria for the evaluation.
- To complete the evaluation, which includes the tasks of: (1) reviewing the results of the evaluation, (2) creating the evaluation report, (3) reviewing the quality assessment, and (4) providing feedback to the organization.
- To provide data during the evaluation process.

III. STUDY CASE: QUALITY EVALUATION PROCESS FOR THE SOFTWARE PRODUCT

This part of the paper explains the details of the execution of the activities indicated in the previous section, in order to implement the quality evaluation process of the email software product. It is important to mention that two email tools will be analyzed i.e. Outlook Web Access (OWA) and Thunderbird.

A. Establishment of Evaluation Requirements

The first step of the evaluation process is to establish the requirements of the software product quality evaluation.

1) Establishment of the purpose of the evaluation: The evaluation process is executed to:

- Understand and assess the positive and negative effects of the product when it is in use.
- Ensure the quality of the software product.
- Measure the software from technical and functional perspective.
- Provide information security.

In our case, it is important to indicate that quality plays a key role when deciding on possible mail solutions.

2) Obtaining software product's quality requirements: The quality requirements of the software product are specified using the generic quality model indicated in ISO/IEC 25010 [10]. For our case, to obtain the requirements of the new mail client, surveys were conducted to the users of the central and regional offices of the organization nationwide.

3) Identification of parts of the product to be included in the evaluation process: In accordance with the quality life cycle, the parts of the software product included in the evaluation were [11]:

- Product specification
- Product during its execution
- Test results

B. Specification of the Evaluation

1) Selection of evaluation modules: The metrics used to evaluate the internal and external quality of the software product were: (1) functional Suitability, (2) reliability, (3) performance efficiency, (4) usability, (5) security, (6) compatibility, (7) maintainability, and (8) portability; and the metrics used for the quality in use of the software product were: (1) effectiveness, (2) efficiency, (3) satisfaction, (4) freedom of risk, and (5) context coverage [12].

2) Definition of the decision criteria for the OWA and Thunderbird metrics: This part defines the decision criteria for the selected metrics. These are thresholds, represented by H (High), M (Medium), L (Low) and N/A (Not applicable), indicates the importance level of the quality requirements (see Table I) [13].

TABLE I. *METRIC DECISION CRITERIA*

Importance	Symbol	Meaning
High	H	It must be evaluated
Medium	M	It can be evaluated (not a must)
Low	L	It will not be evaluated
Not applicable	N/A	Not applicable

3) Establishment of the decision criteria for the evaluation: In this activity, the criteria for the different characteristics evaluated from the sub-characteristics and quality metrics are defined. These results allow the evaluation of the quality of the software product in a general way at a higher level of abstraction (see table II) [14].

TABLE II. *EVALUATION DECISION CRITERIA*

Measurement Scale	Punctuation Levels	Level of Satisfaction
7.50 – 10	Exceeds the requirements	Very satisfactory
5.00 – 7.49	Meets the requirements	Satisfactory
2.51 – 4.99	Minimally acceptable	Unsatisfactory
0 – 2.50	Not acceptable	

C. Evaluation Design

In this activity, a quality evaluation matrix was made that allowed defining the evaluation items to reach the conclusion of which email tool is the best option to use in the Institution (see Table III). Additionally, the planning of evaluation activities were executed.

D. Evaluation Execution

In this activity, the evaluation activities are executed by obtaining the quality metrics and applying the evaluation criteria.

1) Measurements: To perform the measurements in the quality of the software, laboratory tests and pilot tests in the use of the OWA and Thunderbird were carried out in the Technology Department. The applications quality was measured by the Technology Department staff using the criteria indicated in Section B.

2) Application of the decision criteria: The decision criteria are applied based on the values obtained in the measurement of the product.

3) Application of the decision criteria of the evaluation: In this last task, the decision criteria are applied at the level of quality characteristics and sub-characteristics, producing as a result the assessment of the degree to which the software product meets the established quality requirements (see table IV and V).

TABLE III. *QUALITY MATRIX*

Columns	Description
Characteristic	Characteristic name according to the standard
Sub-characteristic	Sub-characteristic name according to the standard
Metrics	Metric name according to the standard
Specific characteristic	Functionalities tested by users
Technical characteristic	Functionalities tested by technical users
State	If a metric is applied or not
Score	Value over 10 according Table VI
Importance Level	Level weighted by the evaluator
Importance Percentage	Percentage weighted according to the evaluator
Final Value of each Characteristic	Partial value calculated per characteristic
Total of each Characteristic	Sum of partial values of each characteristic

TABLE IV. *RESULTS OF QUALITY ANALYSIS APPLIED TO OWA*

Quality	System's Quality	Score Level	Level of Satisfaction
Internal and External	7,29	Exceeds the requirements	Very satisfactory
Usage	8,5	Exceeds the requirements	Very satisfactory
Total	7,9	Exceeds the requirements	Very satisfactory

TABLE V. *RESULTS OF QUALITY ANALYSIS APPLIED TO THUNDERBIRD*

Quality	System's Quality	Score Level	Level of Satisfaction
Internal and External	7,82	Exceeds the requirements	Very satisfactory
Usage	8,8	Exceeds the requirements	Very satisfactory
Total	8,3	Exceeds the requirements	Very satisfactory

E. *Evaluation Conclusion*

Continuing with the stages described in the standard, this chapter concludes the evaluation by analyzing the obtained results.

1) Evaluation Results Review: Figure 4 shows how the internal and external quality of the mail client OWA has a value of 7.29 over 10 (representing 73%) and the quality in use has a value of 8.5 over 10 (representing 85%). The same figure shows the final quality result of OWA with a value of 7.89 over 10 (representing 79%), indicating a satisfactory fulfillment of the requirements of the users. On the other hand, Figure 5 shows how the internal and external quality of the mail client Thunderbird has a value of 7.82 over 10 (representing 78%) and the quality in use has a value of 8.8 over 10 (representing 88%) . The same figure shows the final quality result of Thunderbird with a value of 8.31 over 10 (representing 83%) indicating a satisfactory fulfillment of the requirements of the users.

2) Results obtained from the internal, external and in-use quality evaluation of the OWA mail client: Figure 6 shows the result obtained from the internal and external quality characteristics of the OWA email client. The figure shows that the most important characteristics are Security and Portability. These characteristics stand out since OWA manages Security and Portability schemes such as access control, passwords in the data file, operation in different operating systems and coexistence with several applications.

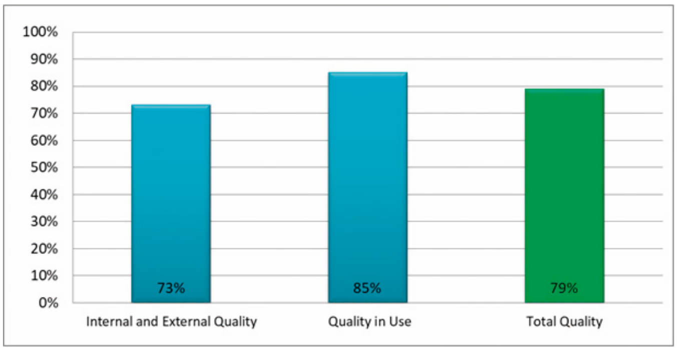

Fig. 4. OWA's Quality Results

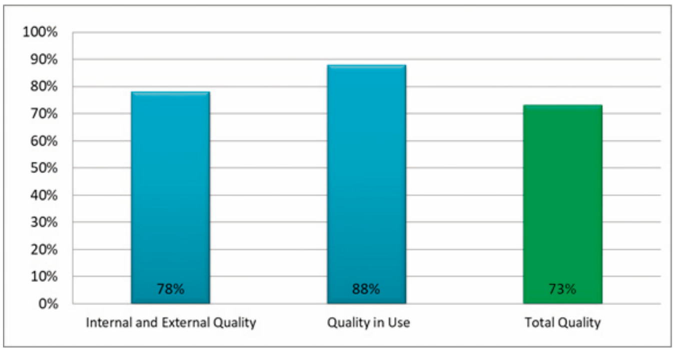

Fig. 5. Thunderbird's Quality Result

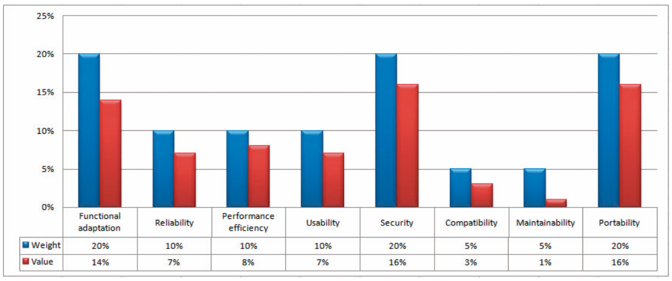

Fig. 6. OWA's Evaluation from Internal and External Quality Perspective

Other characteristics that stand out are the Satisfaction and Freedom of risk. This is because OWA is very similar to Microsoft Outlook which is very familiar to the users of the organization (see Fig. 7)

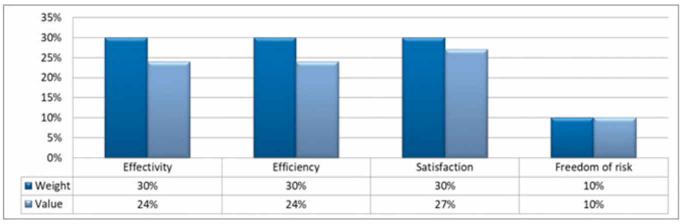

Fig. 7. OWA's Evaluation from Quality in Use Perspective

3) Results obtained from the internal, external and in-use quality evaluation of the Thunderbird mail client: Figure 8 shows the result obtained from the internal and external quality characteristics of the Thunderbird email client. In such figure, it can be evidenced that the most important characteristics are Functional Suitability, Reliability, Usability and Compatibility. These features stand out because Thunderbird maintains an intuitive interface which makes it easy, simple and reliable when sending and receiving emails, and because it is compatible with different software products.

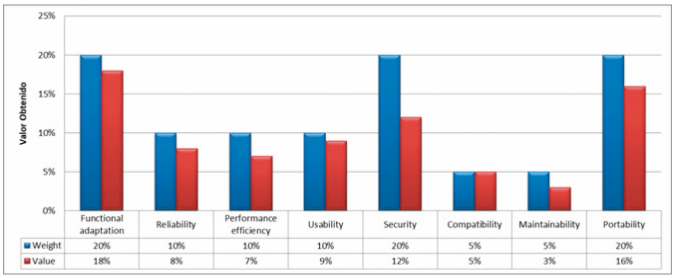

Fig. 8. Thunderbird's Evaluation from Internal and External Quality Perspective

Other characteristics that stand out are the Satisfaction and Freedom of Risk, represented in the percentages detailed below (see Figure 9).

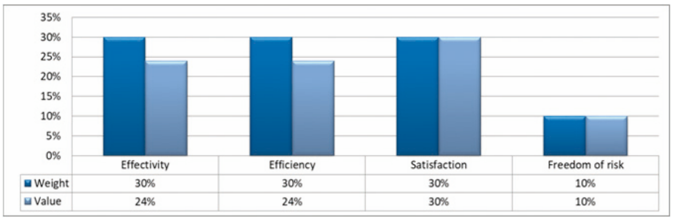

Fig. 9. Thunderbird's Evaluation from Quality in Use Perspective

IV. CONCLUSIONS AND FUTURE WORK

The evaluation models indicated in INEN NTE-ISO/IEC 25000 Standard, which covers the requirements specification and software quality evaluation processes, were able to establish the characteristics and sub-characteristics of software quality and custom metrics to the reality of the organization. Through the characteristics, sub-characteristics and quality metrics obtained in the development of the evaluation model, the internal, external and in-use quality assessment matrix was developed, which considers the results of the tests carried out on the candidate email tools, allowing to reach a single technical solution.

Once the objectives of the presented paper have been reached, interesting research lines are opened that can be used for future work using the NTE INEN-ISO/IEC 25000 standard. One of them could be to define a document that establishes the Ecuadorian governmental model for the evaluation of software products used in Public Sector Institutions based on Agreement 166 and Standard NTE ISO/IEC 27001, having as a purpose of standardizing internal, external and use quality parameters, as well as the final report that should be approved by the pertinent authorities and inform the control organisms.

Another relevant work could be the design, development and implementation of an information system that automates the model developed in this article, which allow each institution to parameterize the system according to its own characteristics and software needs.

Even though the present work was executed for a Governmental organization, it can be applied in any kind of organizations.

In future works, we are planning to compare the applied approach with other methodologies to understand its benefits and limitations.

REFERENCES

[1] ISO/IEC, International Standardization Organization ISO/IEC 25000 Software and Engineering-Software product Quality requirements and Evaluation (SquaRE) – Guide to SquaRE., International Organization for Standarization, ISO 25000, Ginebra, Suiza, 2005.

[2] N. Fenton, S. Pfleeger, Software Metrics: A Rigorous Approach, Chapman, 2010.

[3] ISO/IEC, ISO/IEC 25000. SQuaRE - System and Software Quality Requirements and Evaluation, 2004.

[4] ISO/IEC, ISO/IEC Standard 9126 Software Product Evaluation–Quality Characteristics and Guidelines for their Use 2001, 2001

[5] ISO/IEC, ISO/IEC 14598-5:1998 - Information technology -- Software product evaluation -- Part 5: Process for evaluators. 1998, 1998

[6] ISO/IEC ISO/IEC 25000, Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) --Guide to SQuaRE.

[7] ISO. (2005a). ISO/IEC 25000 software and system engineering – software product quality requirements and evaluation (SQuaRE) –Guide to SQuaRE.

[8] ISO/IEC FDIS 25000 - Systems and Software Engineering -- Systems and Software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, (2005i).

[9] ISO, ISO/IEC 25040 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Evaluation process. 2011, International Organization for Standardization: Ginebra, Suiza.

[10] ISO, ISO/IEC 25010, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. 2011: Ginebra, Suiza.

[11] ISO/IEC 25020:2007 - Software Engineering -- Software Product Quality Requirements and Evaluation (SQuaRE) -- Measurement Reference Model and Guide, (2005d).

[12] ISO/IEC 25021:2012 - Systems and Software Engineering -- Systems and Software Quality Requirements and Evaluation (SQuaRE) -- Quality Measure Elements, (2005e).

[13] ISO, ISO/IEC DIS 25023. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality. 2015.

[14] ISO/IEC TR 25060:2010 - Systems and Software Engineering -- Systems and Software Product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for Usability: General Framework for Usability-Related Information, (2005k).

[15] Suryn W., Abran A., Bourque P., Laporte C., "Software Product Quality Measurement and Evaluation: An integrated life cycle quality model using TL9000 and ISO/IEC 9126." Submitted to JISBD 2003 - VIII Jornadas de Ingeniería del Software y Bases de Datos 12-14 November 2003, Alicante, Spain.

[16] H. Yang, "Measuring Software Product Quality with ISO Standards Base on Fuzzy Logic Technique," Affiective Computing and Intelligent Interaction, pp. 59-67, 2012

[17] B. Russo, E. Damiani, S. Hissam, B. Lundell, G. Succi, "Open Source Development, Communities and Quality," OSS: IFIP International Conference on Open Source Systems, Milano, Italy, September 7-10, 2008

[18] R. Plosch *et al*., "The EMISQ Method - Expert Based Evaluation of Internal Software Quality," *31st IEEE Software Engineering Workshop (SEW 2007)*, Columbia, MD, 2007, pp. 99-108. doi: 10.1109/SEW.2007.71

[19] A. Duque-Ramos et al., "OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies," Journal of Research and Practice in Information Technology, Vol 43, Issue 2, May 2011.