



SOFTWARE QUALITY EVALUATION: A BIBLIOMETRIC ANALYSIS AND FUTURE PERSPECTIVES

Tulio Cremonini Entringer

Universidade Estadual do Norte Fluminense Darcy Ribeiro, Brazil

E-mail: tulio_entringer@hotmail.com

Ailton da Silva Ferreira

Universidade Federal Fluminense, Brazil

E-mail: ailtonsilvaferreira@yahoo.com.br

Submission: 11/23/2018

Revision: 2/08/2019

Accept: 2/27/2019

ABSTRACT

The theme of “Software Quality Evaluation”, and its importance, concerns the standardization of the development processes, in order to guarantee a higher quality of the systems. In this scenario, this theme has gained importance in academic research, due to the emphasis that the theme has increasingly presented to organizations, since they understand that the main objective is to guarantee a final product that satisfies the expectations of the client. In order to verify the scientific scenario in recent years, this work aims to carry out a review of the literature on “Software Quality Evaluation”, based on a bibliometric analysis and review. For this, the method of bibliographic revision with four different phases proposed by Marasco (2008) was used. From the data extracted from the Scopus database, the results of the systematic review and bibliometric analysis were analyzed from the information on publications, citations and approaches of the papers that deal with this subject. As results, the present article showed the academic and practical importance of the studies on “Software Quality Evaluation”. In addition, it is possible to identify gaps in the scientific literature that can be filled by future work in the area, based on the studies developed by specialists.

Keywords: Software; Quality; Software Quality Evaluation; Bibliometric Analysis



1. INTRODUCTION

With the accelerated pace of dissemination of information, and the need for organizations to keep abreast of current events, information becomes increasingly important, because with access to it, organizations can analyze the market and design objectives, decisions based on more credible information.

For the processing of the vast amount of information available in the globalized world, it is important that organizations enjoy the resources and tools of systems and software that have the ability to process information and make it useful.

The role of softwares in the information production process is to receive, process and disseminate important information for the management and control of the activities of an organization. Among the main purposes of an information system is to assist its users in activities, processes and decision making.

The evolution of technology has significantly impacted these systems, since they have been developed with robust and complex computational principles. As a consequence of this impact, organizations started to give more importance to the origin, data processing and quality of the information generated.

Software is important throughout the organization management process. To make these systems efficient, it is important that there is a follow-up, thus verifying if the desired specificities are actually achieved.

Despite the obvious importance, some systems may not achieve their goals. This can happen for several reasons, which can be caused by the lack of operational knowledge on the part of the end user, until the system is not meeting the real needs proposed.

In order to verify the scientific scenario in recent years, this work had the objective of a review of the literature on the evaluation of software quality from a bibliometric analysis.

Several studies have addressed the evaluation of software quality, for example: in telecommunications industry systems (SURYN; LAVERDIÈRE, 2007), in web applications and B2B e-commerce applications (BEHKAMAL; KAHANI; AKBARI, 2009); in ERP projects (PARTHASARATHY; SHARMA, 2017), in systems of the logistics sector (KURTEL; OZEMRE, 2013), among others.



The first section presents the contextualization and the objective of this research. The second section presents an approach to evaluating the quality of software. The third section presents the research methods, which included the systematic bibliographic review and the methods of analysis of publications, citations and approaches. The following section presents the research results, which include the authors and journals that published the most in the study area (publications analysis); the main approaches of articles related to software quality (approach analysis); and the most cited articles in the database searched, taking into account the impact factor of the journal in which the work was published. Finally, the last section presents the findings of the study.

2. SOFTWARE QUALITY EVALUATION

The software is made based on a standardization of the development processes, in order to guarantee a higher quality of the systems. The models applied in quality assurance mainly act in the process, but the main objective is to guarantee a final product that satisfies the client's expectations (GUERRA; COLOMBO, 2009).

The parameters and criteria for the quality of information systems are defined by NBR ISO / IEC 9126-1 as "the totality of the characteristics of a software product that gives it the ability to satisfy explicit and implicit needs" (ABNT, 2003).

The needs expressed in the definition of the requirements to be fulfilled by the system are named of explicit needs, among them we can mention the objective of the system, the functions and the expected performance. Tsukumo (1997) conceptualize implicit needs, as those that are even not expressly stated, are necessary to the system user. The authors emphasize that generally in this class are included the requirements that are not explicit because they are obvious, and also the requirements that were not perceived as necessary in the development of the system.

For Rothery (1995), "quality is appropriateness to use. It is the conformity of the requirements ". Therefore, the author affirms that the quality management involves a priori three elements, they are: definition of the objectives; standards, and a system.

Weinberg (1993) concludes that quality is defined from the perceived value to some person, so the quality is relative, that is, what is quality for one person, may not be for another. For the author, such a perception of quality is always influenced by the politics and emotion of the people involved. Côrtes (2008) adds that "the quality of a

product is subjective and varies according to the place and time ...", where the same product may present higher or lower quality when evaluated by different people.

The main quality indicator in any product, including software, is end-user satisfaction, that is, the parameter that serves to express the value and utility of the system, and how much it meets the user's real needs (INTHURN, 2001).

In order to better satisfy the quality perception of users of a given system, Laudon and Laudon (2011) argue that organizations can improve the quality of their software in two ways: using quality assurance techniques or improving data quality.

Although quality is a factor to be considered in the development of the systems, Tsukumo (1997) state that quality assessment has emerged as a need to improve existing systems, so the need to evaluate this quality is necessary when the product is already in use.

According to Côrtes (2008), the last decades were of high commitment to obtain results in the area of quality, all supported by elaborations of systems of normalization, metrology and certification. This author clarifies these systems, being: (i) standardization the activity that establishes dispositions destined to the common and repetitive use to reach a high degree of order; (ii) metric is the set of operations whose purpose is to determine the value of a quantity; and (iii) certification to prove and declare that a company has products, processes or services in accordance with certain standards.

With increasing concern about quality, companies started to adopt procedures to guarantee the security and quality of the information produced, since information systems, especially those in computer format, are vulnerable to the most varied problems, such as errors, loss and various difficulties such as hardware or software failures (LAUDON; LAUDON, 2011).

Evaluating the quality of systems becomes more and more important, given that users excel more and more at good quality. For Guerra and Colombo (2009), the quality of software should be evaluated at all stages, that is, during development, after the product generated, and finally in the product in use. Although the literature suggests this form of evaluation, the objective of this study is to evaluate the quality of the stage of use of the system, that is, in the step quality measures in use.

The International Organization for Standardization (ISO) listed in NBR ISO/IEC 9126-1, the set of characteristics that must be verified in a software to be considered quality, that is, if it has the characteristics capable of satisfying the users' needs (ABNT, 2003).

According to Khaddaj and Horgan (2004), quality is a multidisciplinary idea reflected in a quality model, where each parameter used in the model refers to a dimension of quality.

There are currently several criteria for assessing the quality of a system, however, organizations may also create their own criteria based on the importance they present to the organization (INTHURN, 2001). Considering that the work in question refers to the end user's point of view of a system, the evaluation criteria were based on NBR ISO/IEC 9126-1. According to NBR ISO/IEC 9126-1, it can be stated that the attributes of software quality can be defined by six characteristics: functionality, reliability, usability, efficiency, maintainability and portability (ABNT, 2003).

3. METHODOLOGICAL RESOURCES

Bibliometrics provides statistical metrics related to the study of quantitative processes of production, dissemination and use of information and also designates advanced processes and mechanisms of online search and information retrieval techniques, being important tools for management of teaching and knowledge, once that it is possible to detect gaps in the scientific literature (BUFREM; PRATES, 2006).

This research can be considered theoretical and conceptual and is focused on the bibliographic analysis to be carried out through a structured review of the literature, aiming to analyze the software quality evaluation.

Will be used the method of bibliographic revision, proposed by Marasco (2008), with four different phases (definition of database, definition of research keywords, selection and analysis of papers), as shown in Figure 1.

In the first phase for the composition of the bibliographic portfolio, the Scopus databases were defined.

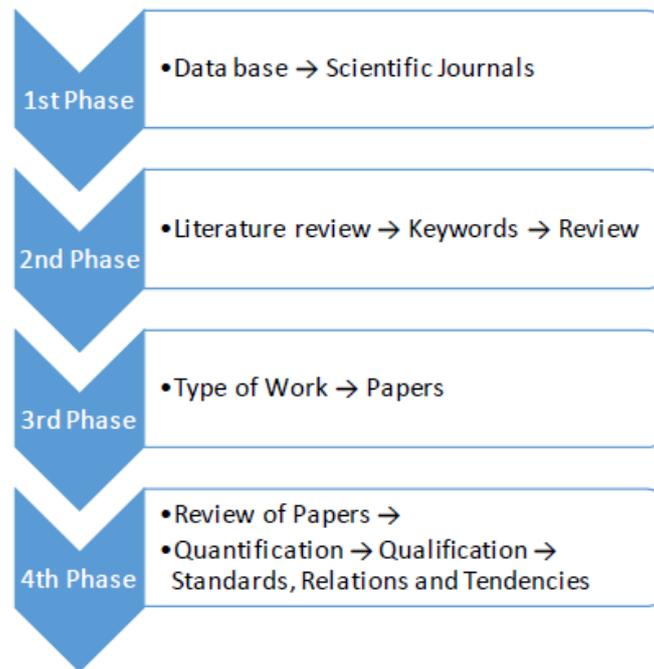


Figure 1: Flow chart of methodology to be applied in the bibliometric review.
Source: author (2018).

Although the results and conclusions from this research are limited to the selected database, the ideas presented in this paper contribute to the development of the field of knowledge and can serve as inspiration for the development of new knowledge, both for academics and professionals in the field of software quality evaluation.

Surveys were conducted in journals without temporal delimitation of published studies. The total amount of works found in the mentioned databases was a result of the combinations of keywords searched in the titles and in the summaries of the papers. The research was conducted in November of the year 2018.

In the phase of definition of keywords for the bibliographic review, from the Scopus database, the following keywords were used: "Evaluation of Software Quality" OR "Software Quality Evaluation".

These terms were submitted to the selection filter, which included the inclusion criterion by reading the title, summary and keywords. This search resulted in a sample of 235 papers published. The selection filter was applied related to the type of document, where only articles were selected, since only these are evaluated by pairs in their final version, resulting in 85 papers.

Subsequent to the definition of the sample, the data available in the Scopus database were extracted as: authors, title, journal, year and number of citations. In the analysis of citations of papers collected at the Scopus database, two activities are presented: calculation of the corrected index of citations and ordering of the most cited papers.

Equation 1 shows the calculation of the Corrected Citation Index (CCi), in which the CI is the citation index extracted from the database and IF is the impact factor of the journal in which the paper was published (IRITANI et al., 2015). According to Lopes and Carvalho (2012), the correction of citation index aims to contain, in addition to the number of citations, the relevance of the journal in which the paper was published.

$$CCi = CI \times (IF + 1) \quad (1)$$

4. RESULTS AND DISCUSSION

The bibliometry analysis is able to discover how the characteristics of publications of the field and, with this, to visualize the scientific scene, the main authors and works, the most relevant journals and the distinctive features that integrated the studies carried out, which helps in the elaboration of dissertations and papers, since it is possible to analyze the existing gaps in the scientific literature of a given area of research.

The first analysis of the publications was of periodicals per year, in which it was possible to identify the evolution of the publications of articles focused on the study on the evaluation of software quality.

Figure 2 shows the evolution chart of the number of publications over the years from ordo to the Scopus database. It is observed that, in the last 35 years, there has been little publication of work on this subject, varying from 0 to 2 publications per year. Between 2001 and 2009, there was a small increase in publications, with a variation of 1 to 4 publications per year. As of 2010, there was another growth of publications, especially in 2011, when there were 9 publications of articles on the evaluation of software quality. Between 2015 and 2018, the average number of publications per year was 4 articles.

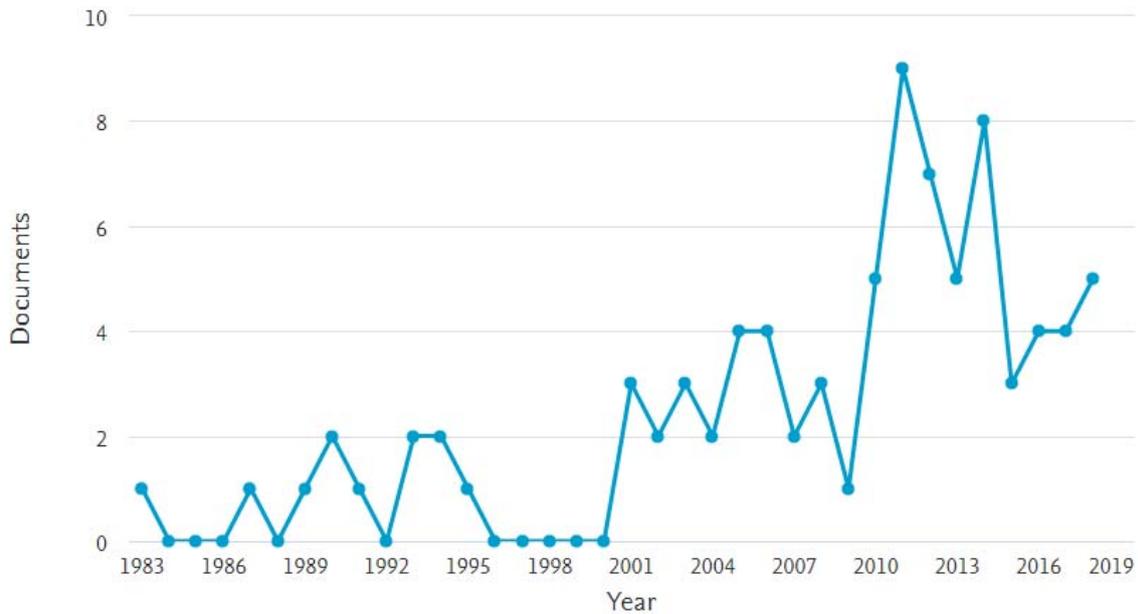


Figure 2: Evolution of the number of publications over the years in the Scopus database.

Source: author (2018).

It is also important to note that in the year 2018 only five studies were conducted focused on this topic. The following are the most recent work in the areas of software quality assessment are:

- Shrestha et al. (2018) used three International Standards (process assessment ISO/IEC 15504-330xx series, IT Service Management ISO/IEC 20000 series, and System and Software Quality Models from ISO/IEC 25010) to present an account of a DSR (Design Science Research) project with three evaluation sites where the artefact was tested in real organizational contexts. The project demonstrated the significant role of International Standards to confirm research relevance during artefact design, development and evaluation.
- Kostin and Smirnov (2018) proposed a generalized model for evaluation of software functionality. The authors also provided a functional evaluation model for machine translation systems developed based on it and that can be a prototype for more detailed models. In this work, we consider some ways to use these models to numerically evaluate the functionality of automatic translation systems.
- Yue (2018) propose a new method to compare two intervals based on two-dimensional geometric interpretation. Equivalence is illustrated by a test.

Some mathematical experiments are shown to illustrate validity. The proposed method is applied to the evaluation of software product quality. The results show that the method provided has important implications in theoretical and practical perspectives.

- Liu et al. (2018) identified three critical elements of a typical crowdsourcing software project, including mass developers, task editors, and crowdsourcing service platform, and proposed a new value-rate quality assessment strategy to evaluate these elements respectively. When assessing the atomic unity of crowdsourcing the quality of crowdsourcing software, the model not only considers the values of the individual units, but also takes into account the factors of the participants, provides a method of calculating the value rate, as a quality metric for a crowdsourcing software project. The authors applied the proposed evaluation strategy on crowdsourcing service platforms and conducted simulation experiments to analyze their effectiveness. The results showed that, with the increase in the number of tasks, a consistent coupling relationship between the value of the contractor and the probability of the user's ability. The proposed model can be extended to model the various factors, besides the value rate, that must be modeled to evaluate a crowdsourcing project.
- Hong and Kim (2018) aimed to understand and recognize which quality assessment factors should be considered by the convergence software used in the main technologies of the 4th revolution of the sector and its importance. The scale of quality evaluation proposed in this study can be applied to software of various application environments, such as drones, autonomous vehicles and artificial intelligence. In assessing the quality of the software, the authors hope that it will be possible to develop a better quality for the software by analyzing the expected big data processing in the software of the application environment through filtered analysis.

Among the 158 authors surveyed, the authors who published the most are show in Figure 3.



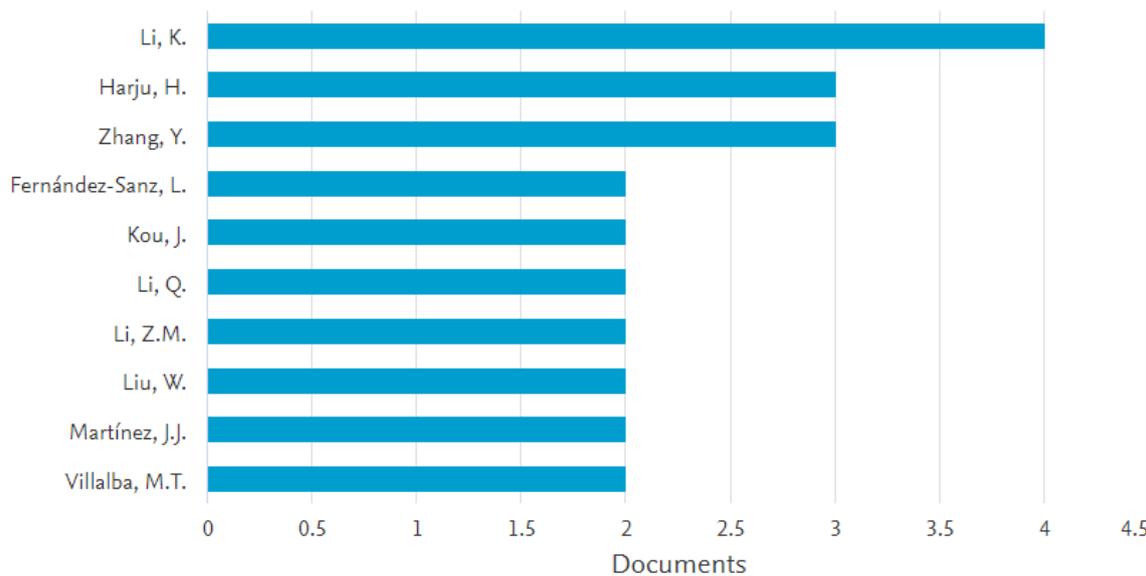


Figure 3: Quantitative publications of the main authors in the Scopus database.

Source: author (2018).

Li, K. stands out for the author with the highest number of publications on the evaluation of software quality, with 4 articles published. Then, with 3 articles published each, Harju, H. and Zhang, Y. appear, and the latter has published together with Li, K. They also stand out because they have 2 publications each: Fernández-Sanz, L. ; Kou, J. ; Li, Q. ; Li, Z. M. ; Liu, W. ; Martínez, J. J. and Villalba, M. T. Then, according to the Scopus database, the articles published by the three authors with the largest number of papers in the study area are:

- Li, Zhang, and Liu (2014) presented an illustrated example to show the application of the proposed model of quality software evaluation, and the results shown that the weights of product characteristics and the customers needs in HoQ (House of Quality) analyzed can make software quality more accurate and comprehensive in software quality evaluation.
- Li, Zhang, Liu, and Gao (2012) presented a fuzzy regression model with trapezoidal fuzzy numbers to express the relationships between the user requirements and software features which can not only meet the fuzziness of the user requirements but meet the complexity of the software products in reality. An illustrated example is presented to show the application of the proposed model, and the results indicate the model is more flexible and lays a good foundation for analyzing the software products quality quantitatively.

- Li, Zhang, and Kou (2012) presented a fuzzy regression model with asymmetric triangular fuzzy numbers to express the functional relationships in order to guide us to adjust development strategy to produce high-quality software. An illustrated example is presented to show the application of the proposed model.
- According to Li, Kou, and Gong (2011), the software quality prediction model is the key technology in the software quality assessment system, which can be used to evaluate the software quality characteristics that users care about. For the authors, predictive models are often used to find the non-linear relationship between metric data and quality factors.
- Rosqvist, Koskela and Harju (2003) developed of a generic and operationally feasible measurement technique to transform the tacit knowledge of a software expert to a probability distribution depicting his/her uncertainty of the level of achievement related to a quality attribute. The authors developed rules for the construction of a consensus probability measure based on expert-specific probability measures and derived a framework for specifying software quality strategy and for evaluating the acceptance of a software produced in a software development process; The above technical developments are used to support group decision-making regarding the launch or implementation decision of a software version; the allocation of resources during the software development process.
- Harju (2002) addressed a brief review of some widely referenced standards, to consider a context for using the approach for other attributes of reliability rather than security. According the author, software quality assessment consists of three parts: personal, product and process evaluations. All of them are also required for a reliability assessment. The publication is focused on product certification from three different points of view: reliability score, software reliability engineering, risk management process.

Table 1 shows the list of extracted, in descending order of the number of citations, at the Scopus database, specifying the number of citations and the that was published, impact fator, according to the Journal Citation Reports (JCR), and the citation index corrected.

Table 1: List of papers and their respective CI, %CI, JIF and CIC.

Reference	Journal	CI	%CI	JIF	CIC
(TANG; FARN, 2005)	Journal of Business Ethics	35	5,76%	2,917	137,095
(MAROTO; TORMOS, 1994)	International Transactions in Operational Research	32	5,26%	2,4	108,8
(LI; ZHANG; LIU, 2014)	Journal of Intelligent and Fuzzy Systems	39	6,41%	1,426	94,614
(CHANG; WU; LIN, 2008)	Software Quality Journal	35	5,76%	1,596	90,86
(ROSQVIST; KOSKELA; HARJU, 2003)	Software Quality Journal	34	5,59%	1,596	88,264
(NIE; LEUNG, 2011)	ACM Transactions on Software Engineering and Methodology	29	4,77%	1,946	85,434
(MIYOSHI; AZUMA, 1993)	IEEE Transactions on Software Engineering	16	2,63%	3,331	69,296
(FONTANA; BRAIONE; ZANONI, 2012)	Journal of Object Technology	64	10,53%	-	64
(PEDRYCZ et al., 2001)	Neurocomputing	15	2,47%	3,241	63,615
(BLIN; TSOUKIÁS, 2001)	Software Quality Journal	16	2,63%	1,596	41,536
(DUQUE-RAMOS et al., 2011)	Journal of Research and Practice in Information Technology	41	6,74%	-	41
(JOHNSTON, 1987)	Journal of Computer Assisted Learning	14	2,30%	1,859	40,026
(RUAN; YANG, 2014)	Mathematical Problems in Engineering	12	1,97%	1,145	25,74
(SIMÃO; BELCHIOR, 2003)	Lecture Notes in Computer Science	18	2,96%	-	18
(KURILOVAS; SERIKOVIENE, 2010)	Informatics in Education	13	2,14%	-	13

CI – Citation Index; %CI – Relative Citation Index; JIF- Journals Impact Factor (2017); CIC – Citation index corrected. Source: authors (2018).

After calculating the citation index correction, the work developed by Tang and Farn (2005) stands out for having the largest CIC number. Next, the article by Chang, Wu and Lin (2008); Li; Zhang and Liu (2014) and Maroto and Tormos (1994) are highlighted.

Tang and Farn (2005) investigated the effect of interpersonal influence on personal software piracy, also known as softlifting. A laboratory experiment with 54 subjects was conducted, in which each subject was told to participate in a software quality evaluation exercise. However, a ploy was carried out to measure the subjects' intention in software piracy under different levels of group pressure and financial gains. A path analysis demonstrated that normative influence was related to softlifting intention, yet information influence effect was marginal. The effect of normative influence on softlifting behavior was mainly mediated by softlifting intention.

Maroto and Tormos (1994) presented a comparative study of a representative sample of packages: CA-Superproject, Instaplan, Micro Planner, Microsoft Project and

Project Scheduler, through the evaluation of quality features such as user-friendliness, documentation, resource allocation.

Li et al. (2014) investigated the multiple attribute decision making (MADM) problem for software quality evaluation based on the geometric aggregation operators with hesitant fuzzy uncertain linguistic information. Then, motivated by the ideal of traditional geometric operation, the authors have developed some geometric aggregation operators for aggregating hesitant fuzzy uncertain linguistic information. The prominent characteristic of these proposed operators are studied. Then, they have utilized these operators to develop some approaches to solve the hesitant fuzzy uncertain linguistic multiple attribute decision making problems. Finally, a practical example for software quality evaluation was given to verify the developed approach and to demonstrate its practicality and effectiveness

Chang et al. (2008) proposed a software quality evaluation model and its computing algorithm with the main objective to propose a novel Analytic Hierarchy Process (AHP) approach for addressing uncertainty and imprecision in service evaluation during pre-negotiation stages, where comparative judgments of decision makers are represented as fuzzy triangular numbers. A new fuzzy prioritization method, which derives crisp priorities from consistent and inconsistent fuzzy comparison matrices, was proposed. The Fuzzy Analytic Hierarchy Process (FAHP)-based decision-making method can provide decision makers or buyers with a valuable guideline for evaluating software quality. Importantly, the proposed model can aid users and developers in assessing software quality, making it highly applicable for academic and commercial purposes.

The works of Fontana, Braione, and Zanoni (2012) and Duque-Ramos, Fernández-Breis, Stevens, & Aussenac-Gilles (2011) stand out because they have more citations, according to a Scopus database, according to the number of journals because they were not considered relevant by a JCR.

This paper of Fontana, Braione, and Zanoni (2012) reviews the current overview of the tools for automatic code smell detection. It defines research questions about the consistency of their responses, their ability to expose the regions of code most affected by structural decay, and the relevance of their responses with respect to future software evolution. It gives answers to them by analyzing the output of four

representative code smell detectors applied to six different versions of GanttProject, an open source system written in Java. The results of these experiments cast light on what current code smell detection tools are capable of and what the relevant areas for further improvement are.

Duque-Ramos, Fernández-Breis, Stevens, and Aussenac-Gilles (2011) proposed a framework for evaluating the quality of ontologies based on the SQuaRE standard for software quality evaluation. This method requires the definition of both a quality model and quality metrics for evaluating the quality of the ontology. The quality model was divided into a series of quality dimensions or characteristics, such as structure or functional adequacy, which are organized into subcharacteristics, such as cohesion or tangledness. Thus, each subcharacteristic was evaluated by applying a series of quality metrics, which are automatically measured. Finally, each characteristic was evaluated by combining values of its subcharacteristics. This work also includes the application of this framework for the evaluation of ontologies in two application domains.

5. CONCLUSION

This paper presents a bibliometric analysis about the quality evaluation applied in software with the objective of identifying the frequency of scientific publications on the subject.

It is observed that, in the last 35 years, there has been little publication of work on this subject, varying from 0 to 2 publications per year. Between 2001 and 2009, there was a small increase in publications, with a variation of 1 to 4 publications per year. As of 2010, there was another growth of publications, especially in 2011, when there were 9 publications of articles on the evaluation of software quality. Between 2015 and 2018, the average number of publications per year was 4 articles.

Li, K. stands out for the author with the highest number of publications on the evaluation of software quality. The following authors also highlighted the number of publications in this area: Harju, H., Zhang, Y., Fernández-Sanz, L., Kou, J., Li, Q., Li, Z. M., Liu, W., Martínez, J.J. and Villalba, M.T.

The results obtained from the citations analysis, which shows the relationship of current and relevant articles in the field, can help in the composition of the bibliographic portfolio in the elaboration of future works in the area of software quality,



since this analysis takes into account, in addition to the number of citations of the articles, the relevance of the periodical in which the article was published. After calculating the citation index correction, the work developed by Tang and Farn (2005) stands out for having the largest CIC number. Next, the article by Maroto and Tormos (1994), Li et al. (2014) and Chang et al. (2008) are highlighted.

It is considered that this article reached its objective, since it presented the scenario of publications about the use of norms and techniques applied in quality of software and its relevance at the moment. This work can be used as support for future researches that seek to use techniques and norms of applied quality in software or motivate researches to look for new alternatives of quality control models applied in software development processes.

REFERENCES

ABNT. (2003) **Engenharia de software - Qualidade de produto. Parte 1: Modelo de qualidade**. Associação Brasileira de Normas Técnicas.

BEHKAMAL, B.; KAHANI, M.; AKBARI, M. K. (2009) Customizing ISO 9126 quality model for evaluation of B2B applications. **Information and Software Technology**, v. 51, n. 3, p. 599–609.

BLIN, M. J.; TSOUKIÁS, A. (2001) Multi-Criteria Methodology Contribution to the Software Quality Evaluation. **Software Quality Journal**, v. 9, n. 2, p. 113–132.

BUFREM, L. S.; PRATES, Y. (2006) O saber científico registrado e as práticas de mensuração da informação. **Ciência da Informação**, v. 34, n. 2.

CHANG, C.-W.; WU, C.-R.; LIN, H.-L. (2008) Integrating fuzzy theory and hierarchy concepts to evaluate software quality. **Software Quality Journal**, v. 16, n. 2, p. 263–276.

CÔRTEZ, P. L. (2008) **Administração de sistemas de informação**. São Paulo: Saraiva.

DUQUE-RAMOS, A. et al. (2011) OQuaRE: A square-based approach for evaluating the quality of ontologies. **Journal of Research and Practice in Information Technology**, v. 43, n. 2, p. 159–176.

FONTANA, A. F.; BRAIONE, P.; ZANONI, M. (2012) Automatic detection of bad smells in code: An experimental assessment. **The Journal of Object Technology**, v. 11, n. 2, p. 5.

GUERRA, A.; COLOMBO, R. M. T. (2009) **Qualidade de produto de software**. Ministério da Ciência e Tecnologia, , 2009. Available: http://www.mct.gov.br/upd_blob/0203/203505.pdf. Access: 11/09/2018.

HARJU, H. (2002) Cost-effective design and assessment of dependable software: Part 1. **VTT Tiedotteita - Valtion Teknillinen Tutkimuskeskus**, n. 2151, p. 3–114.

HONG, J. K.; KIM, J. A. (2018) Case study of convergence software considering



Aesthetics in center of conversion software quality in the 4th industry revolution environment. **Journal of Engineering and Applied Sciences**, v. 13, n. 4, p. 3853–3859.

INTHURN, C. (2001) **Qualidade & teste de software**. Florianópolis: Visual Books.

IRITANI, D. R. et al. (2015) Análise sobre os conceitos e práticas de Gestão por Processos: revisão sistemática e bibliometria. **Gestão & Produção**, v. 22, n. 1, p. 164–180.

JOHNSTON, V. M. (1987) The evaluation of microcomputer programs: an area of debate. **Journal of Computer Assisted Learning**, v. 3, n. 1, p. 40–50.

KHADDAJ, S.; HORGAN, G. (2004) The Evaluation of Software Quality Factors in Very Large Information Systems. **Electronic Journal of Information Systems Evaluation**, v. 7, n. 1, p. 43–48.

KOSTIN, A. V.; SMIRNOV, V. V. (2018) Functionality Evaluation Model for Machine Translation Systems. **Journal of Computer and Systems Sciences International**, v. 57, n. 1, p. 157–169.

KURILOVAS, E.; SERIKOVIENE, S. (2010) Learning Content and Software Evaluation and Personalisation Problems. **Informatics in Education**, v. 9, n. 1, p. 91–114.

KURTEL, K.; OZEMRE, M. (2013) Cohesive software measurement planning framework using ISO standards: a case study from logistics service sector. **Journal of Software: Evolution and Process**, v. 25, n. 7, p. 663–679.

LAUDON, K. C.; LAUDON, J. P. (2011) **Sistemas de informação gerenciais**. São Paulo: Pearson Prentice Hall.

LI, K. et al. (2012) The application of fuzzy regression based on the trapezoidal fuzzy numbers to the software quality evaluation. **Journal of Convergence Information Technology**, v. 7, n. 19, p. 293–300.

LI, K.; KOU, J.; GONG, L. (2011) Predicting Software Quality by Optimized BP Network Based on PSO. **Journal of Computers**, v. 6, n. 1.

LI, K.; ZHANG, Y.; KOU, J. (2012) Fuzzy regression with asymmetric fuzzy coefficients applied to software quality evaluation. **Information**, v. 15, n. 10, p. 3957–3967.

LI, K.; ZHANG, Y.; LIU, W. (2014) Weight Analysis based on ANP and QFD in Software Quality Evaluation. **Applied Mathematics & Information Sciences**, v. 8, n. 2, p. 793–798.

LIU, A. et al. (2018) An effective value-rate model for crowdsourcing software quality evaluation. **Journal of Advanced Oxidation Technologies**, v. 21, n. 2.

LOPES, A. P. V. B. V.; CARVALHO, M. M. DE. (2012) The evolution of the literature on innovation in cooperative relationships: a bibliometric study for the last two decades. **Gestão & Produção**, v. 19, n. 1, p. 203–217.

MARASCO, A. (2008) Third-party logistics: A literature review. **International Journal of Production Economics**, Research and Applications in E-Commerce and Third-Party Logistics Management. v. 113, n. 1, p. 127–147.

MAROTO, C.; TORMOS, P. (1994) Project management: an evaluation of software quality. **International Transactions in Operational Research**, v. 1, n. 2, p. 209–221.

MIYOSHI, T.; AZUMA, M. (1993) An empirical study of evaluating software development environment quality. **IEEE Transactions on Software Engineering**, v. 19, n. 5, p. 425–435.

NIE, C.; LEUNG, H. (2011) The Minimal Failure-Causing Schema of Combinatorial Testing. **ACM Transactions on Software Engineering and Methodology**, v. 20, n. 4, p. 1–38.

PARTHASARATHY, S.; SHARMA, S. (2017) Impact of customization over software quality in ERP projects: an empirical study. **Software Quality Journal**, v. 25, n. 2, p. 581–598.

PEDRYCZ, W. et al. (2001) Calibration of software quality: Fuzzy neural and rough neural computing approaches. **Neurocomputing**, v. 36, n. 1–4, p. 149–170.

ROSQVIST, T.; KOSKELA, M.; HARJU, H. (2003) Software quality evaluation based on expert judgement. *Software Quality Journal*. v. 11, n. 1, p. 39–55.

ROTHERY, B. (1995) **ISO 14000 and ISO 9000**. Brookfield, Vt: Gower.

RUAN, C.; YANG, J. (2014) Software Quality Evaluation Model Based on Weighted Mutation Rate Correction Incompletion G1 Combination Weights. **Mathematical Problems in Engineering**, v. 2014, p. 1–9.

SHRESTHA, A. et al. (2018) Benefits and relevance of International Standards in a design science research project for process assessments. **Computer Standards & Interfaces**, v. 60, p. 48–56.

SIMÃO, R. P. S.; BELCHIOR, A. D. (2003) Quality characteristics for software components: Hierarchy and quality guide. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 2693, p. 184–206.

SURYN, W.; LAVERDIÈRE, F. (2007) The applicability analysis of ISO/IEC 9126 series of standards in telecom sector: the example of softswitch category of software. **Software Quality Journal**, v. 15, n. 4, p. 417–448.

TANG, J.-H.; FARN, C.-K. (2005) The Effect of Interpersonal Influence on Softlifting Intention and Behaviour. **Journal of Business Ethics**, v. 56, n. 2, p. 149–161.

TSUKUMO, A. N. (1997) Qualidade de software: visões de produto e processo de software. In: II ESCOLA REGIONAL DE INFORMÁTICA DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO REGIONAL DE SÃO PAULO. Piracicaba: **Proceedings...**Piracicaba: SBCRSP, 1997.

WEINBERG, G. (1993) **Software com qualidade - Pensando e idealizando sistemas**. Makron, v. 1.

YUE, C. (2018) A novel approach to interval comparison and application to software quality evaluation. **Journal of Experimental & Theoretical Artificial Intelligence**, p. 1–20.