A Bibliographic Review of Software Metrics: Applying the Consolidated Meta-Analytic Approach

Ari Melo Mariano, Ana Carla Bittencourt Reis, Lucas dos Santos Althoff and Laís Bandeira Barros

Abstract This research aimed to provide an integrating model for the principle contributions of the scientific literature on software metrics with an impact on that field. Software metrics has emerged as a crucially important aspect of software development planning insofar as indicators representing development costs and the effort involved are essential for the formulation of new systems proposals and the analysis of the systems already in use inside organizations. To that end an exploratory, quantitative, bibliometric study was carried out using the Theory of the Consolidated Meta-analytic Approach. The study retrieved 658 relevant registrations from the Web of Science database for the period 2010–2018. The main contributions and most important approaches are presented together with an integrating model with three main classifications (a) Metrics for Quality in Cloud (40.93%), (b) Software Metrics as Technique (29.30%), and (c) Current Uses of Software metrics (29.77%). In addition the taxonomy of the most cited articles was established and comparisons were made with the results from multi-language databases like Scopus and Google scholar.

Keywords Software metrics · Theory of the consolidated meta-analytic approach · Categorization · Integrating model

A. M. Mariano (✉) · A. C. B. Reis · L. dos S. Althoff · L. B. Barros
Projeto MAP, Department of Production Engineering, Faculty of Technology, University of Brasília, Brasília, DF, Brazil
e-mail: arimariano@unb.br

L. dos S. Althoff
e-mail: ls.althoff@gmail.com

L. B. Barros
e-mail: laisbandeirab@gmail.com

J. Reis et al. (eds.), *Industrial Engineering and Operations Management I*, Springer Proceedings in Mathematics & Statistics 280, https://doi.org/10.1007/978-3-030-14969-7_21

1 Introduction

The increasing applications of Information and Communication Systems means that the most varied organizations devote ever greater efforts to developing software. Output demands and the growing complexity of data call for high levels of processing capacity to meet the requirements of Systems that are increasingly integrated and complex and still has to be elaborated in different programming languages.

In that regard, software metrics emerges as a crucial aspect of software development planning insofar it can work as indicators representing development costs and the effort involved are essential for the formulation of new systems proposals and the analysis of the systems already in use inside organizations. Thus, software measurement is a particularly important procedure in the sense that it provides essential information about the software artefact itself [1].

The theme of Software metrics is a particularly difficult to address due to its internal diversity [2]. In general, metrics is defined as the observable value which results from some measurement assigned to attributes of the real or abstract world. It verifies the size, quantity or degree of some attribute using a validated measuring device. In the computational context there is continual testing and proposal of new quantitative indicators with greater capacity in respect to the respective context.

The importance of this subject stems from the current context of increasing integration of market participants in the course of digital transformation in which software has become the mechanism for consolidating the great variety of information that industries need for their very existence and their development [4.0]. From the social point of view, understanding software metrics literature means having updated indicators of IT systems (i.e. complexity, quality and productivity), collaborating in the progress of software development, enabling improved utilization and ensuring appropriate support for organizations or countries to conduct their planning activities. The present bibliometric review, in particular, will assist the Brazilian Army to identify recent advances in the field of software metrics collaborating with the dimensioning of its developers teams and determining the costs of development systems designed to unify previously unfolded actions improving the effectiveness of their IT systems.

The purpose of this study is to explore where software metrics research is situated as well as the tendencies related to this subject by bibliometric reviewing the main recent papers obtaining answers to the research questions: (a) who are the leading authors? (b) which are the most-used approaches? (c) which are the most important lines of research today?

This paper sets out to provide a model for integrating the main contributions in the scientific literature with an impact on software metrics and to that end it adopts the Theory of the Consolidated Meta-analytic Approach (*Teoria do Enfoque Metaanalítico Consolidado*—TEMAC). To achieve this, the review is organized appropriately around the Methodology (Sect. 2) where we present our research method, Results and Analyses (Sect. 3) which contain the literature review itself with the result of the research, followed by the final remarks (Sect. 4).

2 Methodology

This exploratory study adopts the Theory of the Consolidated Meta-analytic Approach (TEMAC) [3]. It consists of three stages: (1) preparing the research, (2) Presenting and inter-relating data, (3) Detailing the integrating model and evidence-based validation.

The first stage embrace respectively the definition of key-words, the kind of documents, the timeframe of interest, the databases to be consulted and the area of knowledge the review will investigate. In the second stage the bibliometric information is extracted from the literature databases and then the laws of bibliometrics are applied in an analysis of the relations among the extracted data. An analysis of those articles that have historically been the most cited makes use of CitNetExplorer software to portray the evolution of contributions to the theme.

Lastly integrating and validating models will be applied to the evidence obtained from the Citation, Bibliographic Coupling and Co-occurrence mapping study. Those steps are based on Pritchard's laws of bibliometrics [4]. The importance of using bibliometric techniques is underscored by the increasingly rapid generation of scientific-technological information, corresponding to an ever increasing number of publications and broadening the scope of the task of identifying the most important ones and their interconnections [5].

This study used the 1.6.5 version of the *VOSviewer* software to ensure a satisfactory analysis of the bibliometric data. Heat maps concentrate similar information in clusters and a color scale indicates the degree of importance of the respective articles or key-words; red denotes the most important and blue the least important in each one of the bibliographic displays used. The construction of the heat maps to represent interrelations of information made use of the "density" visualization option. That made it possible to combine the cumulative analysis with the network analysis [6].

The survey made use of the string "software metrics" and searched for correspondence in the Web of Science (WoS) database with a time interval filter of 2010–2018. The choice of that particular database was because it is widely acknowledged to be one of the most complete and prestigious of the specialized scientific literature databases [7].

To consolidate more recent data in the integrating model, Iramuteq software was used to achieve a Descendent Hierarchic Classification of sixty abstracts from articles published in 2017 and 2018. In addition, the most cited articles were classified taxonomically after being carefully read.

In advance, we have chosen a single database for the whole analysis for transparency purposes. To ensure a wide outreach for the review the results of WoS were compared with results in multilingual databases, notably Scopus and Google scholar (GS).

3 Results and Analysis

3.1 Preparing the Research

The first stage of TEMAC is preparatory and is orientated by the following questions: (a) What is the research's descriptor, string or key-word? (b) What is the research's time-space field? (c) Which database will be mined? (d) Which areas of knowledge will be delimited?

For this general overview we defined the key-word "software metrics" for the period 2010–2018, using the Web of Science database. The areas of knowledge filtered were: clinical neurology; optics; environmental engineering; mechanical engineering; ocean engineering; energy fuels; environmental sciences; geography; gastroenterology; hepatology; imaging science and photographic technology; physical geography; green sustainable science technology; applied mathematics; logic; nuclear physics; mechanics; applied physics; particle and field physics; biology; neurosciences; nuclear science technology; education scientific disciplines; mathematical computational biology; medicinal chemistry; multidisciplinary chemistry.

The filtered search retrieved 658 registrations. The filter was selected in the light of the Brazilian Army and the Researcher's interest in collecting information associated with organizational context.

For the multilingual comparison it was selected the same key word and time interval for the Scopus research resulting in 1415 documents, and for the GS research that was done through Publish or Perish extracting the first 1000 results. All research was carried out from 8th of February to 15th of March.

3.2 Data Presentation and Inter-relations

The second stage of TEMAC is presenting and inter-relating the data. The data are drawn from the same Web of Science platform and also obtained by actual reading of the articles obtained after the filtering process using the relevance criteria.

The earliest study found is entitled "Exploring the Influence of Identifier Names on Code Quality: an empirical study" by [8] published in the annals of the 14th European Conference on Software Maintenance and Reengineering and it addresses the question of source code quality. This continuation of an earlier work by the same author evaluates source code and identifier quality using software metrics. The results show that poor quality identifiers are associated to low grade source codes. The study was evaluated in a medical context (diagnosis tests) and proved to be satisfactorily consistent.

The average number of citations for the whole set of retrieved articles is 3.06 while the most cited one, "What's up with software metrics? - A preliminary mapping study" by Barbara Kitchenham was cited 70 times. In her work Kitchenham makes a systematic review of the years 2000–2005 and reveals a panorama of tendencies

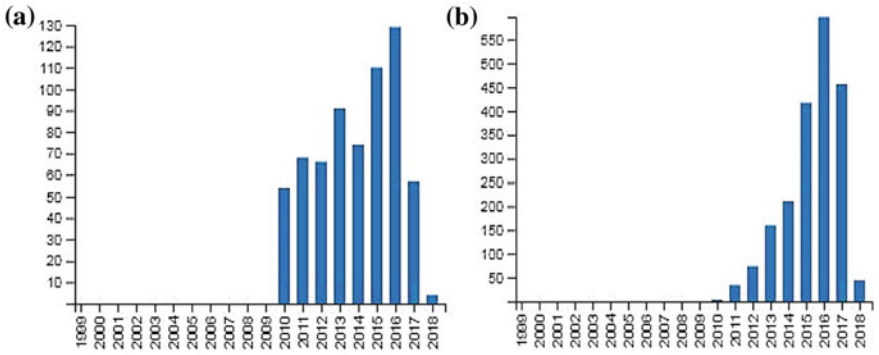

Fig. 1 **a** Evolution of the number of publications. **b** Evolution of the number of citations on the theme of software metrics. *Source* Web of Science

among articles with a strong impact in the field of software metrics. She confirms the wide scope of that area and states how difficult it is to evaluate the current state of research investigating that topic, suggesting that new systematic investigations need to be elaborated capable of categorizing the fields and tendencies associated to the theme.

Most of the 658 studies retrieved by the database search (2010–2018) are proceedings papers (n = 427, 65%) followed by 225 articles (34%) and just 6 reviews (0.9%), one of which was the single most cited work.

In regard to the evolution of citations over time, it can be seen from Fig. 1 that software metrics has been incrementing its notability in the scientific community in spite of the substantial drop in the number of publications in the year 2017 when it went back to the level of 2010.

Among the publications retrieved, the author with the highest number of citations is Khoshgoftaar, T. M., with 155. Works of his that are cited mainly refer to software quality aspects and especially to the correct use of software metrics to forestall defects and avoid risks. Seliya, N. is the second most cited author with 102 citations. That number, however, is largely because he appears as co-author of many of Khoshgoftaar, T. M.'s publications.

In third place comes Williams, L., with 89 citations. His articles present the main approaches to software metrics as well as feedback on the use of metrics and its complexity. Wang, H. J., (87) and Gao, K. H., (85), are in fourth and fifth position respectively also with their numbers boosted by publications in which they appear as co-authors with Khoshgoftaar, T. M., revealing the existence of a robust study nucleus.

Among the articles with the highest numbers of citations (71) is "What's up with software metrics? - A preliminary mapping study", by Kitchenham [2]. It presents work referring to empirical studies of software development and assessment in industries and organizations as, for example, those by authors [9, 10] who discuss the importance of software metrics in decision-making in the context of large

organizations dedicated to software development or in aggregating quality to those metrics. The second most cited work (69 citations) "Software fault prediction metrics: A systematic literature review" by Radjenović [19] is a systematic review of the literature for the period 1991–2011 and it identifies and evaluates software metrics capacity to predict flaws. That author's approach takes into account how the context influences metrics selection and performance. In his results he categorizes metrics in three types: (1) Object-oriented metrics, (2) Source code metrics and (3) Process metrics.

Shin [11] is the author of the third most cited work, ("Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities", with 62 citations. In his article the author creates an empirical model and conducts a case study of two large-scale projects studying indicators for code vulnerability in relation to discriminant and predictive capacity in which 28 metrics divided into three measurement categories are tested in regard to (1) Complexity (2) Code Churn, and (3) Developer activity [11]. "Empirical validation of object-oriented metrics for predicting fault proneness models", by Singh [21] is the fourth most cited work with 60 citations. The author concentrates on metrics software models for predicting flaws or defects which are also associated to studies of indicators for estimated development effort. In regard to Brazilian publications, the outstanding author is Ferreira [22] in tenth position with 36 citations. In his work that author describes object-orientated software and definitions of the thresholds that control the violations and principles of software design.

Having obtained an identification of the main documents in terms of citation, it is important to verify the presence of those authors that publish most in this area and gain an understanding of their influence on the field of software metrics. The most prolific author in publishing terms is Khoshgoftaar, T., with 22 works and he is also the author of the most quoted article ("an empirical study of feature ranking techniques for software quality prediction"). It is an empirical study involving software quality prediction. Part of the published works of Misra, S. (13) and Gao, K. (9) are those in which they appear as co-authors and they are accordingly, the major exponents of the theme. Other authors like Napolitano, A. (12) and Dohi, T. (9) also appear but their publications have lower numbers of citations.

Another factor that suggests how widely studies on the theme of software metrics are disseminated is the variety of countries that produce material in that area. Among those that produce the most are India, the United States, Brazil and China. India has one of the most cited authors, Singh and Rath, S. K. who is responsible for the greatest quantity of Indian publications. It is also worth underscoring Brazil's contribution on that theme and the authors who published most are Garcia, A. (9) and Figueiredo, E. (5).

The Key-words that occurred most frequently in the titles and abstracts are suggestive of the main lines of research. The main key-words are: software (746), metrics (502), prediction (185), design (122), quality (110), object-oriented (107), models (102), defect (100), code (84), analysis (57), and complexity (55). There is a visible tendency towards the use of models designed for defect prediction or that analyze quality or complexity using software metrics with object-orientated codes.

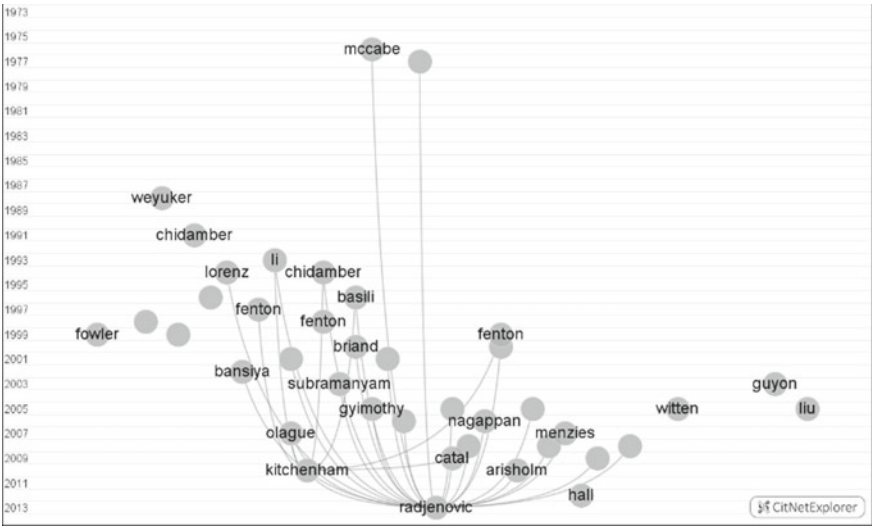

Fig. 2 Map cross-referencing authors' citations as a function of time. *Source* Web of Science. Map generated using CitNetExplorer

A preliminary review of the articles demonstrates conclusively that the lines of research that make most use of metrics software are those designed to predict flaws and defects in object-orientated software. There are also many studies in this area that formulate empirical models that seek to validate software complexity. Finally there is a third line of software metrics studies that focus on the control and management of software development processes.

The last action in this stage is to cross-reference the data on the earlier patterns of citations; that is, referring to other periods prior to 2010–2018 and their evolution over time, together with their authors (Fig. 2).

It can be seen that authors Mccabe, T. J. (1976) and Halstead [20] are the precursors of software metrics research and that their discoveries are still being reflected in more recent studies of authors like Radjenović [19], who stands out as the most robust reference insofar as he carried out a complete review of the literature and identified the most cited authors.

Once the main literature on the theme and the lines of research are identified by analyzing which are the most cited works, key-words and authors and those that published the greatest number of materials and once the evolution of the citations over the course of time has been portrayed, it is time to pass on to the third stage of TEMAC which consists of detailing, applying the integrating model and evidence-based validation.

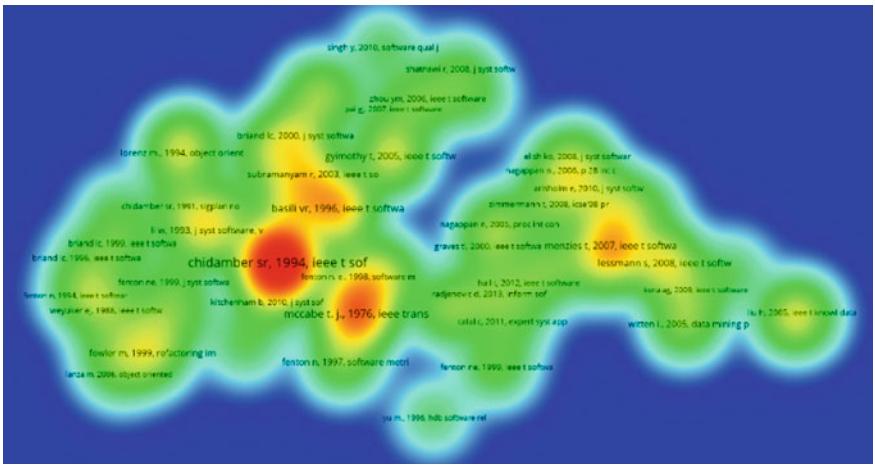

Fig. 3 Co-citation density map *Source* Web of Science. Map generated by VOSViewer

3.3 Detailing, Integrating Model and Evidence-Based Validation

The detailing is achieved using co-citation which is a search for the main approaches and for any coupling, with a view to presenting the principal sources of the research. After that the integrating model will be presented with the evidence-based validation.

Figure 3 shows the density map based on co-citation. Mccabe, T. J. (1976) has an approach nucleus of a historical nature. Based on his work one can see the growth of interest in the complexity of software in 1976. Halstead (1977) appears in that very same nucleus ratifying the earlier work. Those were the origins of software metrics.

There is another red patch in Fig. 3 corresponding to the work of Chidamber [24], who reports on the increasing improvement of processes driving the growth of software metrics, especially for object-orientated projects and that is in agreement with Basili [25], thereby formatting one of the strongest metrics approaches to object-orientated projects. Lastly there is a third pale red blotch in which the work of Menzies, T., appears, revealing a third approach to software metrics involving their application in data mining for the purpose of avoiding errors by means of defect predictors.

With the three most important approaches identified, the next step is to use Bibliographic Coupling (Fig. 4) to map the currently most important sources for research. To correct for the time factor in the number of citations this research considered the normalized numbers of citations [12].

The diversity of approaches in the current literature stands out. There is a miscellany of measuring and evaluation methods and studies of the various different software metrics.

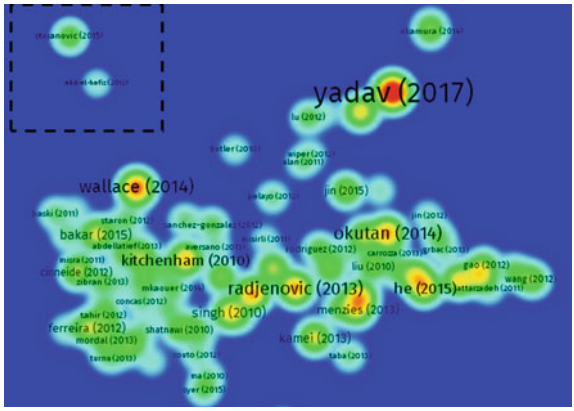

Fig. 4 Bibliographic coupling density map. **a** Elements displaced from the image frame inserted. *Source* Web of Science. Map generated by VOSViewer

In this case there is only one red nucleus visible but there 7 orange ones and 7 yellowish ones. Some of the yellow nuclei are close to other yellow ones suggesting a certain proximity of the respective studies. The single red nucleus refers to Yadav [13] and indicates a model tested and validated in the literature as an indicator that predicts defects of software in the early stages of development (analysis of requirements, design, implementation) offering qualitative information based on fuzzy logic associated to cost saturation, future problems and optimized development strategies [13]. The nucleus represented by Wallace's [14] empirical study proposes and tests a prediction model on the use of software metrics based on the Technology Acceptance Model. Those authors intention is to offer a guide for software engineers when selecting software measurements and to facilitate coordination of the software metrics planning [14]. Another nucleus that stands out is the one represented by Okutan [15] which conducts an empirical study of software metrics related to defect prediction for Bayesian Networks in an effort to optimize the set of metrics used on the basis of an investigation of Promise data repository metrics [15]. Radjenović's [19] work, mentioned above, dialogues with Madeyski [26] and Singh's [21] outstanding empirical studies, both studying flaw prediction; the first using process metrics and the second validating an object-orientated metrics [16]. Shin [11] and Menzies [27] represent a focus on the ways of thinking software engineering data; a factor that should be determinant in metrics selection. Inside another perspective, he [17] argue in favor of process metrics presenting a guide for the selection of a simplified metrics set.

Two analyses were performed with the abstracts of the articles for 2017 and 2018, to create the integrating model. One of the analyses was based on Descending Hierarchic Classification with the aim of determining the main classes that are addressing software metrics; the other was based on an actual reading of the most cited articles, classifying them according to the type pf research and the levels of metrics and characteristics that have intersections, all with the aim of understanding the taxonomy of those most cited articles.

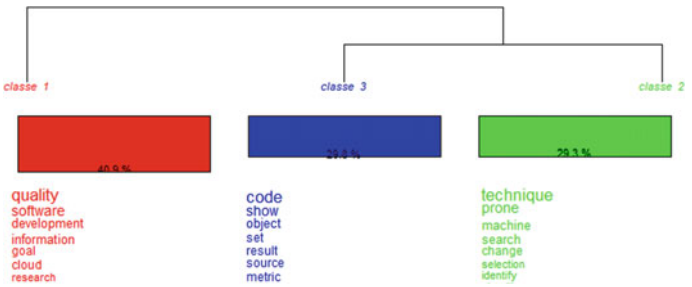

Fig. 5 Descending hierarchic classification dendrogram. Taken from Web of Science. Dendrogram generated in Iramuteq

The first analysis examined 60 abstracts and found 311 text segments of which 70.01% were made use of. The text segments were organized into three classes: Class 1 with 40.93%, Class 2 with 29.30% and Class 3 with 29.77% (Fig. 5).

In Class 1 the most representative works appear, namely: Hussein, A.; Kumar, N.; Okamoto, T.; Tirumalai, S. V.; Dahab, S. A.; Zhao, F.; Ganea, G.; Densumite, S.; Lumpe, M.; Ma, J.; Ali, M. M.; Shi, Y. With the exception of Okamoto, T., all those authors published articles in 2017. Analyzing the words that represent the class such as Quality, Software, Development, Information and Goal, it can be seen that they all share a similar concern for software quality right from its development by means of metrics and especially in cloud computation environments. Accordingly the glass is called **Metrics for Quality in Cloud**. In class 2 the works of Basal, A.; Malhotra, R.; Yan, Y. Q.; Shatnawi, R.; Stuckman, J.; Yohannese, C. W.; Morasca, S.; Azzeh, M.; and Huijens, H., are the most representative. Analyzing the words most associated to class 2 such as Technique, Prone, Machine, Search and Change, it can be seen that the authors' research is directed at software metrics utility such as techniques for constructing research and machine-learning algorithms. There are also some studies based on prediction, statistical tests and comparisons of models to test their efficacy and they too make use of software metrics. According the class is referred to as **Software Metrics as a Technique**. Lastly there is Class 3 where the representative authors are Kumar, L.; Gil, Y.; Anwer, S.; Yadav, H. B.; Kumar, C.; Suganantham, S.; Ebad, S. A.; Cinneide, M. O.; Mansoor, U.; Zhang, F.; Savic, M.; Gu, A. H.; Sultana, K. Z.; and Scalabrino, S. All of them had their articles published in 2017. The most frequently occurring words in that class are Code, Show, Object, Set, Result, Source and Metric. The analysis of those words suggests a strong tendency to studies on the use of object-orientated and source code-orientated software metrics, thereby ratifying their authors' interest in aforementioned themes. That being so the class was called **Current uses of software metrics**.

Thus from the Descending Hierarchy Analysis it can be concluded that the three classes that were found and duly denominated, namely: (a) Metrics for Quality in Cloud, (b) Software Metrics as a Technique, and (c) Current Uses of Software metrics, agglutinate more recent work (2017–2018) addressing software metrics and integrate

Table 1 Identification of the principle authors in the citation method. *Source* Web of Science

Author	Article	No. citations WoS/Scopus/GS index	Category
Kitchenham, B	What's up with software metrics?—a preliminary mapping study	70/103/181	(i)
Radjenović, D	Software fault prediction metrics: a systematic literature review	69/111/186	(i) (1)
Shin, Y	Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities	62/115/204	(ii, iii) (B, C) (2)
Singh, Y	Empirical validation of object-oriented metrics for predicting fault proneness models	60/–/147	(iii) (A) (1, 7)
Gao, K	Choosing software metrics for defect prediction: an investigation on feature selection techniques	52/92/146	(ii) (1, 3, 2)
Menzies, T	Local versus global lessons for defect prediction and effort estimation	41/–/129	(iii) (C) (1, 4)
Ferreira, K	Identifying thresholds for object-oriented software metrics	38/63/100	(iii) (A) (6)
Liu, Y	Evolutionary optimization of software quality modeling with multiple repositories	33/52/81	(ii) (5, 1, 3)
Brown, N	Managing technical debt in software-reliant systems	–/142/245	(iv) (C) (4)
Alves, T	Deriving metric thresholds from benchmark data	–/83/135	(iii) (B) (6)
Rahman, F	How, and why, process metrics are better	45/76/124	(iii) (B, C) (1, 3)
Chowdhury, I	Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities	36/66/131	(iii, iv) (A, B) (1, 2, 4, 7)

them to a three-part integrating model. The second analysis consisted of a meticulous reading of the most cited articles in order to obtain the taxonomy of the most important works (Table 1).

In order to map the amplitude of the presence of the works in databases open to languages other than English, a comparison was made with their citations in Scopus and GS [18].

In regard to types, the studies were organized into (i) Systematic reviews (ii) Case studies (iii) Empirical Models (iv) Conceptual study. As regards the type of metrics the were: (A) Object or design-orientated Metrics (B) Source code metrics (C) Process metrics (D) Traditional metrics (E) Dynamic metrics. In regard to the characteristics of the metrics studied they were: (1) Flaw and defect predictor metrics (2) Software vulnerability indicators (3) Selection of Software quality characteristics (4) Effort and Development estimation (5) Software evolution and quality metrics (6) Threshold metrics (7) Complexity metrics.

Those articles that presented systematic reviews were the most cited but the majority of the studies are of the empirical model type. The works addressing source code and process metrics were the most frequent and most cited together with those addressing object or design-orientated Software Metrics. These results corroborate the classes found earlier in the more recent works. Lastly, as regards the most frequently registered characteristics in the most cited studies they are the flaws and defects predictor metrics followed by the Software vulnerability indicators.

The three levels of categorization were found to be independent and what characterizes the measures is considered to be the objective of the author in exploring a given metrics, one which might be considered by other authors as being merely an indicator for another different measure. Therefore, the so-called characteristics of the measurement can be considered in sub-categories of indicators that partially represent those characteristics. The finality of the empirical studies is to certify, statistically, to what extent a given indicator predicts specific categories. In order to maintain consistency, however, our categorization did not attain that level of detail

It can be seen that some of the work that is indexed in the *Web of Science* (WoS) is not in the Scopus and vice versa. Furthermore, there is a greater number of citations registered in the database that has the wider linguistic scope and above all in the GS which is open.

According to the thresholds established in [3], the theme of software metrics has proved to be based on solid evidence because the use them is the object of more than one systematic review of literature on well delineated, randomized experiments [2, 19], in addition various research centers around the world are studying the theme. Accordingly, not only the principle contributions regarding software metrics but also the importance of the review studies have achieved a great number of citations and aroused considerable interest, as witness the works of Kitchenham [2] and Radjenović [19]. A general analysis of the results obtained shows Radjenović [19] at the center of the various maps, ratifying that author as an indispensable source for understanding the theme.

4 Final Remarks

This research aimed to provide a model to integrate the principle contributions of the scientific literature on software metrics with an impact on that field, in obedience to the steps established by a bibliometric methodology of an exploratory nature

denominated the consolidated meta-analytic approach theory. The integrating model revealed three classes named in accordance to the macro-context of the studies set: (a) Metrics for Quality in Cloud (40.93%), (b) Software Metrics as Technique (29.30%), (c) Current Uses of Software metrics (29.77%). The study identified the main authors, most common approaches and principle lines of research established according to the laws of bibliometrics thereby providing a response to research questions.

It is believed that the results will assist the Brazilian Army to identify the most up to date and appropriate software metrics to apply in its IT systems program planning [14]. As an agenda for the future it is hoped to expand the class studies to all research samples and to be able to present a panorama of changes, period by period. As well as including analysis from other relevant databases (e.g. GS, Scopus, IEEExplorer, Springer Link etc.) in order to absorb all relevant works related to the subject.

References

1. Voulgaropoulou, S., Spanos, G.: Analyzing measurements of the R statistical open source software (2012). ieeexplore.ieee.org
2. Kitchenham, B.: What's up with software metrics?—a preliminary mapping study. J. Syst. Softw. **83**(1), 37–51 (2010)
3. Melo, A., Br, M.A.: Revisão da Literatura: Apresentação de uma Abordagem Integradora. In: AEDEM International Conference, pp. 427–443 (2017)
4. Pritchard, A.: Statistical bibliography or bibliometrics. researchgate.net
5. Gurzki, H., Woisetschläger, D.M.: Mapping the luxury research landscape: a bibliometric citation analysis. J. Bus. Res. **77**, 147–166 (2017)
6. Zupic, I. Čater, T.: Bibliometric methods in management and organization. Organ. Res. Methods **18**(3), 429–472 (2015)
7. Mariano, A.M., García, R., Arenas-Gaitan, J.: Meta analysis as a tool of research: a systematic review of bibliography applied study of international strategic alliances. An. ADMPG, no. August (2011)
8. Butler, S., Wermelinger, M., Yu, Y., Sharp, H.: Exploring the influence of identifier names on code quality: an empirical study. In: 2010 14th European Conference on Software Maintenance and Reengineering, pp. 156–165 (2010)
9. Staron, M.: Critical role of measures in decision processes: managerial and technical measures in the context of large software development organizations. Inf. Softw. Technol. **54**(8), 887–899 (2012)
10. Mordal, K., Anquetil, N., Laval, J., Serebrenik, A., Vasilescu, B., Ducasse, S.: Software quality metrics aggregation in industry. J. Softw. Evol. Process **25**(10), 1117–1135 (2013)
11. Shin, Y., Meneely, A., Williams, L., Osborne, J.A.: Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities. IEEE Trans. Softw. Eng. **37**(6), 772–787 (2011)
12. Van Eck, N.J., Waltman, L.: VOSviewer Manual. 1 January 2013, no. January, 1–28 (2013)
13. Yadav, H.B., Yadav, D.K.: Early software reliability analysis using reliability relevant software metrics. Int. J. Syst. Assur. Eng. Manag. **8**(S4), 2097–2108 (2017)
14. Wallace, L.G., Sheetz, S.D.: The adoption of software measures: a technology acceptance model (TAM) perspective. Inf. Manag. **51**(2), 249–259 (2014)
15. Okutan, A., Yıldız, O.T.: Software defect prediction using Bayesian networks. Empir. Softw. Eng. **19**(1), 154–181 (2014)
16. Abdellatief, M., Sultan, A.B.M., Ghani, A.A.A., Jabar, M.A.: A mapping study to investigate component-based software system metrics. J. Syst. Softw. **86**(3), 587–603 (2013)

17. He, P., Li, B., Liu, X., Chen, J., Ma, Y.: An empirical study on software defect prediction with a simplified metric set. Inf. Softw. Technol. **59**, 170–190 (2015)
18. Harzing, A.-W., Alakangas, S.: Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison. Scientometrics **106**(2), 87–804 (2016)
19. Radjenović, D., Heričko, M., Torkar, R., Živkovič, A.: Software fault prediction metrics: a systematic literature review. Inf. Softw. Technol. **55**(8), 1397–1418 (2013)
20. Halstead, M.H.: Elements of Software Science, vol. 7. Elsevier, New York (1977)
21. Singh, Y., Kaur, A., Malhotra, R.: Empirical validation of object-oriented metrics for predicting fault proneness models. Softw. Qual. J. **18**(1), 3–35 (2010)
22. Ferreira, K.A.M., Bigonha, M.A.S., Bigonha, R.S., Mendes, L.F.O., Almeida, H.C.: Identifying thresholds for object-oriented software metrics. J. Syst. Softw. **85**(2), 244–257 (2012)
23. McCabe, T.J.: A complexity measure. IEEE Trans. Softw. Eng. **SE-2**, 308–320 (1976)
24. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Trans. Softw. Eng. **20**(6), 476–493 (1994)
25. Basili, V.R., Briand, L.C., Melo, W.L.: A validation of object-oriented design metrics as quality indicators. IEEE Trans. Softw. Eng. **22**(10), 751–761 (1996)
26. Madeyski, L., Jureczko, M.: Which process metrics can significantly improve defect prediction models? An empirical study. Softw Qual. J. **23**(3), 393–422 (2015)
27. Menzies, T. et al.: Local versus global lessons for defect prediction and effort estimation. IEEE Trans. Softw. Eng. **39**(6), 822–834 (2013)