A Review of Approaches for Quality Model Validations in the Context of Cloud-native Applications

Robin Lichtenthäler and Guido Wirtz

Distributed Systems Group, University of Bamberg, Germany
{robin.lichtenthaeler,guido.wirtz}@uni-bamberg.de

Abstract. Quality models considering internal design characteristics of software should represent reality as accurately as possible. This can be ensured through a validation of relations between quality attributes. In this work we review validation approaches used in literature. We conclude that in an early design phase, surveys and expert interviews are suitable to validate quality attributes and their relations while for complete quality models quantitative validations through measures are advised.

Keywords: Quality model, Empirical validation, Cloud-native

1 Introduction

Quality models are used in software engineering to enable a structured assessment of the quality of a system according to quality attributes deemed important for it [12]. To do so, quality models typically include a theory of how certain software characteristics are related to higher level quality attributes and how such characteristics and attributes can be measured and combined to enable a quantitative quality assessment [1]. An example would be the theory used by Bansiya et al. [4] that in an object-oriented system the characteristic of coupling impacts extendability in the sense that high coupling has a negative impact on extendability. Coupling is stated to be quantitatively measurable by counting for each class to how many other classes it is directly related [4]. Such theories therefore constitute the inner basis of a quality model and the degree to which they are able to represent the reality ultimately determines the applicability and usefulness of a quality model. AL-Badareen et al. [1], however, also state that quality models are often formulated in a subjective manner and refer to a finding from Kitchenham and Pfleeger [25] that "*software quality models suffer from a lack of rationale for the relationships between quality characteristics and how the lowest levels properties are composed into an overall assessment of higher level quality characteristics*" [1]. The question of how quality models can be validated, that means how well they represent reality, is therefore relevant and important for providing quality models that are useful in practice.

In a recent study [33], we have formulated a quality model for cloud-native application architectures that is based on the Quamoco quality meta-model

J. Manner, D. Lübke, S. Haarmann, S. Kolb, N. Herzberg, O. Kopp (Eds.): 14th ZEUS Workshop, ZEUS 2022, Bamberg, held virtually due to Covid-19 pandemic, Germany, 24–25 February 2022, published at http://ceur-ws.org

[48]. Quality attributes are called *factors* in the Quamoco context and they can be either higher-level *quality aspects* or lower-level measurable *product factors*. To formulate *factors* and their interrelationships, called *impacts*, we relied on the ISO 25010 standard [19] in combination with suggestions from practitioner books. Nevertheless, the resulting quality model includes a subjective notion. A validation of the model and especially of the stated impacts would be beneficial for future work that uses the model. In this work, our aim therefore is to review existing approaches for validating quality models, especially in an empirical way. Our quality model for cloud-native applications [33] serves as a use case for which we want to derive implications for how a validation of such a newly created quality model could be performed and which aspects need to be considered. The contribution of this work is an overview of how and how often quality models proposed in literature are validated. In addition, we review approaches that exist to ensure the validity of quality models with implications for the formulation of new quality models. To summarize this, we aim to answer the following research questions:

RQ1: To what extent and how are quality models proposed in literature validated?

RQ2: Which implications can be derived for a proper validation of newly formulated quality models?

In the following, we provide some foundations on quality models in Sect. 2, discuss related work in Sect. 3 and present our methodology in Sect. 4. We describe our results and answers to our research questions in Sect. 5, before concluding our work with an outlook in Sect. 6.

2 Hierarchical Software Quality Models

The term quality model is to some extent used ambiguously and can for example also refer to a list of rules checked through static code analysis [34] where quality is measured directly based on the number of rule violations found in a software. In this work, however, the focus is on so-called hierarchical quality models [4] where a hierarchy exists from lower-level measures to higher-level quality aspects. Hierarchical quality models can integrate and interrelate multiple quality aspects and enable a more detailed evaluation of software quality. In turn, theories are needed to state the relationships between lower-level measures and higher-level quality aspects mediated by software characteristics.

In broad fields, such as software engineering, theories are difficult to generalize and often apply well only within certain contexts. Therefore, different quality models exist for different domains, for example object-oriented systems [4], embedded systems [35], Web services [42], or SOA architectures [16]. A contrast to these specializations are the emerged standards for quality in software: ISO 9126 [18] and its successor ISO 25010 [19]. The consequence, however, is that these standards mainly cover higher level quality attributes and advice on how to measure and evaluate software quality. Nevertheless, the standards provide a theoretical basis which has been validated through the structured definition and

refinement process involving a group of experts. But for context-specific quality models including lower level quality attributes and measures the question of how well the underlying theory maps to reality remains. The methodological approach for ensuring the validity of a theory, and therefore of a quality model, is referred to as validation. We distinguish validation from evaluation in this work by considering evaluation as the approach of using a quality model for evaluating the quality of a software. Although in literature, these two terms are used inconsistently. Another distinction which is important for discussing different validation approaches, is that between *internal characteristics* of a software which are evaluated by analyzing the internal implementation of a system and *external characteristics* of a software which can only be evaluated at runtime when observing its behavior. This distinction is in line with the ISO 9126 [18] and ISO 25010 [19] standards which consider internal and external quality, or Kitchenham et al. [26] who differentiate between "externally visible properties" and "internally visible properties". For both types of characteristics impacts on quality aspects can be stated, but typically impacts of external characteristics are more intuitive. An example would be the externally visible uptime of a system for which it can be stated that a high uptime has a positive impact on the availability quality aspect. Especially considering internal characteristics, a clear rationale for the relationships between quality attributes is therefore important, as also stated by Wagner et al. [48] who developed Quamoco, a meta-model for quality models, in which relationships between quality attributes are defined as *impacts*. When formulating a quality model based on their meta-model such impacts need to be stated based on valid reasoning, which can for example rely on logical reasoning, previous literature, or empirical methods where empirical evidence is considered to be statements from people with experience in the domain of a quality model, collected through interviews or surveys. Empirical methods in specific are also found important for the acceptance of quality models in practice by Moody [39] who reviewed quality attributes of conceptual models (which is a superset of models used in software engineering and therefore also includes quality models). In conclusion, that means that also a validation of a hierarchical software quality model should put a focus on the validity of the stated impacts for a quality model which describe the relationships between the different factors, also considering their importance in relation to each other. On the basis of these properties of hierarchical quality models, we designed our approach for reviewing quality models and validation approaches applied to them.

3 Related Work

Our work is related to work considering the quality assurance of quality models themselves. In contrast to our perspective on the validity of the underlying theory, quality models can also be evaluated based on structural aspects: AL-Badareen et al. formulate a set of rules [1] for structural aspects of the impact graph of quality attributes. Furthermore, they formulate a set of rules for quality characteristics construction which, however, take into consideration a specific system to be

evaluated. In addition, Moody [39] has done an evaluation of quality models for conceptual models and argues that empirical validation is important. He also discusses different methods to do so (e.g., laboratory experiments, action research, or surveys), but with a focus on the validation through applying a quality model, which is also important, but not practicable during the early design phase of a quality model as it is the case with our quality model for cloud-native applications. In addition, previous work has also systematically reviewed different quality models taking validation into consideration, but it is only addressed shortly. For example, Nistala et al. [41] only assess whether an evaluation has been done, but it is unclear if explicit validations of quality models are meant or evaluations of software systems using the quality model. Although Nistala et al. also report whether empirical approaches have been used, they are not discussed in detail. Yan et al. [49] shortly address validation methods used for quality models and find the categories *expert opinion*, *issue handling indicators*, and *industry validation*, but do not go into detail except from mentioning that validation is important for practical usage.

4 Methodology

To get an overview of quality models proposed in literature and have a basis for our investigation, we rely on review papers that have already searched the literature for software quality models in a structured way and that reported the quality models they have found as results. As recent review papers, we found the one by Galli et al. [13] (23 results) who aim to measure the relevance of quality models, as well as the systematic mapping studies by Nistala et al. [41] (40 results), focusing on types of model elements used, and by Yan et al. [49] (31 results), focusing on the scope and maturity of quality models. Additionally, we included the mapping study by Oriol et al. [42] (47 results), because of their thematically related focus on web services. Because these review papers present their results in different ways, we hereby introduce the term *entry* to consolidate the results of these review papers in a generic way. An *entry* refers to a research undertaking which may span one or several publications and may or may not explicitly report a quality model. This way we can consider a quality model that has been presented in one paper, but validated, applied, or evolved in additional papers, as a single entry. And we can also include papers that do not explicitly report a quality model, but for example methods for validation which have been proposed independently from a specific quality model. After merging the results from the review papers and removing duplicates, we had 121 entries as an initial set for our investigation. Next, we classified all these entries according to the following criteria:

- **Type of contribution**: Not all studies present hierarchical quality models with explicit factors and impacts, only those that do were classified as contributing a *model*. Others contribute a *meta-model* for quality models, just a *taxonomy* which cannot be used as a quality model, or any kind of

method in the context of quality models (for example how to create a quality model or apply it within an organization). Studies that present a specific method for validating quality models were classified as *validation-method*.

- **Characteristics** considered: For each quality model we differentiate between the characteristics it considers, namely *internal characteristics* and *external characteristics* of a software as described in Sect. 2.
- **Rationale** for non-trivial relationships between quality attributes: This criterion covers the rationale which is used for stating impacts between factors and their relative strength, however considering only non-trivial relationships (In contrast, a trivial relationship would be that a lower latency positively impacts performance efficiency). A rationale can be argumentation simply based on *logical implication* or by relying on existing work (*literature-based*). Empirical evidence can be provided by relying on a small set of experts (*empirical-experts*), for example through interviews, or a structured survey among a larger set of participants (*empirical-survey*) can be used. For a quantification of the relative strengths of impacts, *algorithmic* evaluations are sometimes used. If no rationale is provided or it is not possible to determine it, we classified an entry as *none*.

It has to be noted that for each criterion multiple values could be assigned, for example when both internal and external characteristics are considered or a model together with a method is presented. To ensure that we do not miss validations of quality models published separately after the publication of a quality model, we performed a forward search by looking at the citations for each entry. However, we restricted the forward search to a filtered list of entries which only includes entries where the type of contribution includes a *model* or *evaluation-method* and the considered characteristics include *internal characteristics*. Our focus on internal characteristics is due to our use case of the quality model for cloud-native application architectures [33] which aims to evaluate software architectures at design time based on architectural models. Formulating impacts on quality aspects from internal characteristics is more difficult, because the actual behavior of a system can only be observed at runtime. Therefore validations for such stated impacts are especially important. For the forward search we used SemanticScholar[1] and searched the citations with the keywords **evaluation** or **validation**. The forward search lead to an additional set of 11 publications. Together with the filtered list of entries our final list which forms the basis of our investigation thus consists of 50 entries and can be found online[2]. Detailed information on the used literature, the search process, and classifications can also be found in the corresponding repository for this site[3]. To answer our research questions, we then quantitatively and qualitatively investigated these 50 entries to gain insights and provide implications for validations of quality models.

[1] `https://www.semanticscholar.org/`
[2] `https://r0light.github.io/qualitymodel-validations-review/`
[3] `https://github.com/r0light/qualitymodel-validations-review`

5 Results & Implications

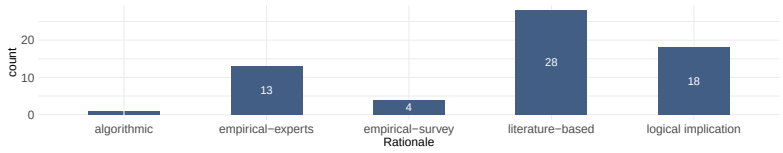

Fig. 1. Count of different types of rationale used

Overall, we were interested in the types of rationale on which the theoretical concepts of quality models are based. In Fig. 1 it can be seen that the majority of quality models relies on previously published literature or uses logical implication to infer conceptual relationships between factors. From a historical perspective it can be seen that early quality models for example from Boehm [5], McCall [37], or Dromey [11] have been formulated and described in comprehensive research reports using experience and logical implication. Later works then relied on them [4, 43] until the ISO standards [18, 19] became available and were again frequently used as a basis [3, 8–10, 14, 17, 21, 22, 24, 28, 32, 40, 44–47]. This shows a tendency to rely on existing literature for a sound theoretical foundation so that a further validation is less important. Empirical approaches are nevertheless also frequently used, especially for more recent domain-specific quality models [15, 35, 36, 38] which are more difficult to cover with the more general standards.

To answer **RQ1** we classified each entry according to whether an explicit approach has been used to validate the proposed quality model. Out of the 50 entries in our result set, 40 do present a quality model and from these we found 18 which included an explicit validation approach. The approach and scope of the validations however are diverse and we therefore further classified the validation approaches according to the scope in focus. For this classification of validations based on their scope we also considered the remaining 10 entries presenting validation methods, independently from a specific quality model An overview of the different validation approaches is provided in Fig. 2. On the left side of Fig. 2 the elements of a quality model in the sense of the Quamoco meta-model are shown while on the right side different perspectives on a software system are shown. The arrows represent relations and the stethoscopes (🩺) signify measures attached to the different perspectives on a system. So, for example, a measure at the source code level could be used to measure the degree to which a product factor is present and the product factor impacts a quality aspect which in turn might impact a higher level quality aspect. The numbered magnifiers (🔎) show the different scopes of validation approaches depending on which elements or relations are in focus. Generally, a differentiation can be done based on the amount of information needed for a validation and the point in time when a validation is suitable. In Table 1 additional details for the different validation approaches are

provided, including which elements of a quality model are required. It can be seen that the validation of factors themselves (V1), the impacts between factors (V2) and relative weights of impacts (V3) can be done solely based on factors proposed for a quality model, and therefore also early in the design phase of a quality model (early in the sense that only factors are defined). Using interviews with experts, Gerpheide et al. [15] have defined and validated factors (V1) and Mayr et al. [35] early validated their model (V1, V2, V3) regarding comprehensibility, appropriate level of abstraction, and consistent classifications through conducting multiple workshops. Lampasona et al. [31] present an approach to rate the minimality and completeness of factors (V1) using interviews with experts. Surveys among practitioners have been used by Mehmood et al. [38] and Gerpheide et al. [15] to validate impacts (V2) and their weights (V3) from product factors on quality aspects by calculating the agreement of respondents regarding the existence and type of impacts. In a similar way, also Khomh et al. [23] validated impacts and their weights (V2, V3), although they investigated design patterns instead of product factors. In our opinion, however, product factors are just a more general construct through which also patterns can be expressed. An additional frequently used approach for assigning weights to impacts on quality aspects (V3) is the Analytical Hierarchical Process [2, 7, 29] in which experts compare impacts for a quality aspect pairwise and based on that weights are calculated.

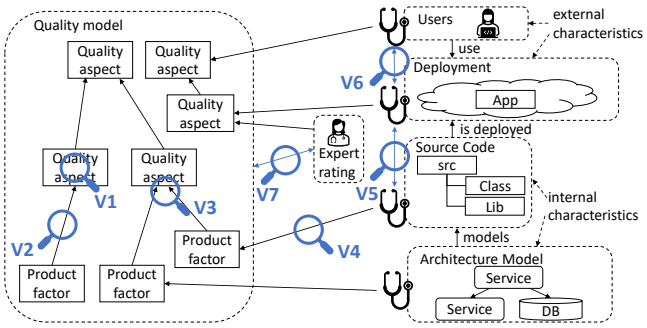

Fig. 2. Overview of possibilities for quality model validations

In contrast to that are validations for completely defined quality models, that means quality models with factors, measures, and relationships between them: Kläs et al. [27] validated factors based on diversification (V1) of measures and overall validity through a comparison with expert ratings (V7). A comparison with expert ratings (V7), also for relations specifically (V3, V4), has been done by Bansiya et al. [4] and Mayr et al. [35]. Braeuer et al. [6] validated measures by comparing them with previously gained measurements (V4). Finally, also considering external measures, Jung et al. [20] compared external measures with user measures (V6) while Kvam et al. [30] and Yu et al. [50] correlated internal measures with external measures (V5), such as productivity or performance.

Table 1. Validation approach scope details

Validation target	Required elements	Examples	Early?
V1 Quality Aspect	Factors	[15, 27, 31, 35]	✓
V2 ProductFactor-QualityAspect	Factors	[15, 23, 35, 38]	✓
V3 ProductFactorImpactWeights	Factors	[2, 7, 15, 23, 29, 35, 38]	✓
V4 CodeMeasure-ProductFactor	Factors, Measures	[4, 6, 35]	
V5 DeploymentMeasure-CodeMeasure	Factors, Measures	[30, 50]	
V6 UserMeasure-DeploymentMeasure	Factors, Measures	[20]	
V7 ExpertRating-QualityModelResult	Factors, Measures, Expert Rating	[4, 27, 35]	

Regarding **RQ2** and the context of our quality model [33] we can therefore state that in an early phase of quality model formulation, where not all elements of a quality model are defined yet, surveys and interviews can be used for validation. This fits the context of cloud-native applications, because it is a comparatively new topic where less existing literature to rely on exists. Therefore, there is also a lack of measures focusing on the architectural level of service interactions and cloud deployment options [33] which makes validations where such measures are needed difficult. Nevertheless, when all elements of a newly formulated quality model are defined, the quality model should also be validated by comparisons of complete evaluations with earlier evaluations or independent evaluations by experts. In addition, it is common to rely on standards as a foundation which can therefore also be recommended for new quality models. A challenge that remains is the large number of factors [33] for which no implications can be derived from the literature, because the considered quality models contained less factors. Finally, an interesting observation is that we did not find any validations for quality models taking architectural models into consideration.

6 Conclusion and Outlook

Ensuring the validity of quality models regarding their internal conceptual basis is important for their applicability and usefulness in practice. During our investigation we found that creators of quality models mostly rely on existing literature for a validated foundation, but also explicit empirical methods are frequently used, especially for domain-specific quality models. A limitation of our work is that we relied on existing survey papers for our literature base, but we added a forward search based on the considered literature to also include more recent work. We plan to apply these results on our recently proposed quality model for cloud-native applications by performing a survey to validate its factors and impacts.

References

1. AL-Badareen, A.B., Desharnais, J.M., Abran, A.: A suite of rules for developing and evaluating software quality models. In: Software Measurement, pp. 1–13. Springer International Publishing (2015), https://doi.org/10.1007/978-3-319-24285-9_1
2. Alrawashdeh, T.A., Muhairat, M.I., Alqatawneh, S.M.: A quantitative evaluation of ERP systems quality model. In: 11th International Conference on Information Technology: New Generations. IEEE (2014), https://doi.org/10.1109/itng.2014.37
3. Alvaro, A., Almeida, E., Meira, R.: Towards a software component quality model. In: Submitted to the 5th International Conference on Quality Software. Citeseer (2005)
4. Bansiya, J., Davis, C.: A hierarchical model for object-oriented design quality assessment. IEEE Transactions on Software Engineering 28(1), 4–17 (2002), https://doi.org/10.1109/32.979986
5. Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative evaluation of software quality. In: Proceedings of the 2nd International Conference on Software Engineering. p. 592–605. ICSE '76, IEEE, Washington, DC, USA (1976), https://doi.org/10.5555/800253.807736
6. Braeuer, J., Ploesch, R., Saft, M.: Measuring maintainability of OO-software - validating the IT-CISQ quality model. In: Advances in Intelligent Systems and Computing, pp. 283–301. Springer International Publishing (2016), https://doi.org/10.1007/978-3-319-46535-7_22
7. Correia, J.P., Kanellopoulos, Y., Visser, J.: A survey-based study of the mapping of system properties to ISO/IEC 9126 maintainability characteristics. In: IEEE International Conference on Software Maintenance. IEEE (2009), https://doi.org/10.1109/icsm.2009.5306346
8. Dixit, D.: Cbqm: Component based quality model. In: 4th International Conference on Reliability, Infocom Technologies and Optimization. pp. 1–5. Citeseer (2013)
9. Domínguez, K., Pérez, M., Grimán, A.C., Ortega, M., Mendoza, L.E.: Software quality model based on software development approaches. In: 11th IASTED International Conference on Software Engineering and Applications. p. 1–6. SEA '07, ACTA (2007), https://doi.org/10.5555/1647636.1647638
10. Dominguez-Mayo, F.J., Escalona, M.J., Mejias, M., Torres, A.H.: A quality model in a quality evaluation framework for MDWE methodologies. In: Fourth International Conference on Research Challenges in Information Science (RCIS). IEEE (2010), https://doi.org/10.1109/rcis.2010.5507323
11. Dromey, R.: A model for software product quality. IEEE Transactions on Software Engineering 21(2), 146–162 (1995), https://doi.org/10.1109/32.345830
12. Ferenc, R., Hegedűs, P., Gyimóthy, T.: Software product quality models. In: Evolving Software Systems, pp. 65–100. Springer Berlin Heidelberg (2013), https://doi.org/10.1007/978-3-642-45398-4_3
13. Galli, T., Chiclana, F., Siewe, F.: Software product quality models, developments, trends, and evaluation. SN Computer Science 1(3) (2020), https://doi.org/10.1007/s42979-020-00140-z
14. Gehlert, A., Metzger, A.: Quality Reference Model for SBA. S-CUBE consortium (2008)
15. Gerpheide, C.M., Schiffelers, R.R.H., Serebrenik, A.: Assessing and improving quality of QVTo model transformations. Software Quality Journal 24(3), 797–834 (2015), https://doi.org/10.1007/s11219-015-9280-8
16. Goeb, A., Lochmann, K.: A software quality model for SOA. In: Proceedings of the 8th international workshop on Software quality - WoSQ '11. ACM (2011), https://doi.org/10.1145/2024587.2024593

17. Hu, W., Loeffler, T., Wegener, J.: Quality model based on ISO/IEC 9126 for internal quality of MATLAB/simulink/stateflow models. In: IEEE International Conference on Industrial Technology. IEEE (2012), https://doi.org/10.1109/icit.2012.6209958
18. ISO/IEC: ISO/IEC 9126 software engineering— product quality. Online (2001), https://www.iso.org/standard/22749.html
19. ISO/IEC: ISO/IEC 25000 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE). Online (2014), https://www.iso.org/standard/64764.html
20. Jung, H.W.: Validating the external quality subcharacteristics of software products according to ISO/IEC 9126. Computer Standards & Interfaces 29(6), 653–661 (2007), https://doi.org/10.1016/j.csi.2007.03.004
21. Kalaimagal, S., Srinivasan, R.: Q'facto 12: an improved quality model for cots components. ACM SIGSOFT Software Engineering Notes 35(2), 1–4 (2010), https://doi.org/10.1145/1734103.1734116
22. Kanellopoulos, Y., Tjortjis, C., Heitlager, I., Visser, J.: Interpretation of source code clusters in terms of the ISO/IEC-9126 maintainability characteristics. In: 12th European Conference on Software Maintenance and Reengineering. IEEE (2008), https://doi.org/10.1109/csmr.2008.4493301
23. Khomh, F., Guéhéneuc, Y.G.: DEQUALITE: building design-based software quality models. In: 15th Conference on Pattern Languages of Programs. ACM (2008), https://doi.org/10.1145/1753196.1753199
24. Kim, C., Lee, K.: Software quality model for consumer electronics product. In: Ninth International Conference on Quality Software. IEEE (2009), https://doi.org/10.1109/qsic.2009.58
25. Kitchenham, B., Pfleeger, S.: Software quality: the elusive target [special issues section]. IEEE Software 13(1), 12–21 (1996), https://doi.org/10.1109/52.476281
26. Kitchenham, B., Linkman, S., Pasquini, A., Nanni, V.: The squid approach to defining a quality model. Software Quality Control 6(3), 211–233 (1997), https://doi.org/10.1023/a:1018516103435
27. Kläs, M., Lochmann, K., Heinemann, L.: Evaluating a quality model for software product assessments – a case study. In: Proc. of SQMB (2011)
28. Kumar, A., Grover, P.S., Kumar, R.: A quantitative evaluation of aspect-oriented software quality model (AOSQUAMO). ACM SIGSOFT Software Engineering Notes 34(5), 1–9 (2009), https://doi.org/10.1145/1598732.1598736
29. Kumar, P., Singh, S.K.: A comprehensive evaluation of aspect-oriented software quality (AOSQ) model using analytic hierarchy process (AHP) technique. In: 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA). IEEE (2016), https://doi.org/10.1109/icaccaf.2016.7748957
30. Kvam, K., Lie, R., Bakkelund, D.: Legacy system exorcism by pareto's principle. In: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications - OOPSLA '05. ACM (2005), https://doi.org/10.1145/1094855.1094959
31. Lampasona, C., Mayr, A., Saft, M.: Early validation of software quality models with respect to minimality and completeness: An empirical analysis. In: Software Metrik Kongress (2013)
32. Lee, K., Lee, S.J.: A quantitative software quality evaluation model for the artifacts of component based development. In: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05). IEEE (2005), https://doi.org/10.1109/snpd-sawn.2005.7

33. Lichtenthäler, R., Wirtz, G.: Towards a Quality Model for Cloud-native Applications. In: European Conference on Service-Oriented and Cloud Computing. ESOCC 2022. Springer (2022), (to appear)
34. Louridas, P.: Static code analysis. IEEE Software 23(4), 58–61 (jul 2006), https://doi.org/10.1109/ms.2006.114
35. Mayr, A., Plosch, R., Klas, M., Lampasona, C., Saft, M.: A comprehensive code-based quality model for embedded systems: Systematic development and validation by industrial projects. In: IEEE 23rd International Symposium on Software Reliability Engineering. IEEE (2012), https://doi.org/10.1109/issre.2012.4
36. Mayr, A., Plosch, R., Saft, M.: Objective measurement of safety in the context of IEC 61508-3. In: 39th Euromicro Conference on Software Engineering and Advanced Applications. IEEE (2013), https://doi.org/10.1109/seaa.2013.32
37. McCall, J.A., Paul K. Richards, G.F.W.: Factors in software quality. volume i. concepts and definitions of software quality. techreport ADA049014 (1977)
38. Mehmood, K., Si-Said Cherfi, S., Comyn-Wattiau, I.: Data Quality through Conceptual Model Quality - Reconciling Researchers and Practitioners through a Customizable Quality Model. In: ICIQ (International Conference on Information Quality), Hasso-Plattner-Institute Potsdam, Germany. pp. 61–74. France (2009)
39. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Engineering 55(3), 243–276 (2005), https://doi.org/10.1016/j.datak.2004.12.005
40. Mordal-Manet, K., Balmas, F., Denier, S., Ducasse, S., Wertz, H., Laval, J., Bellingard, F., Vaillergues, P.: The squale model: A practice-based industrial quality model. In: IEEE International Conference on Software Maintenance. IEEE (2009), https://doi.org/10.1109/icsm.2009.5306381
41. Nistala, P., Nori, K.V., Reddy, R.: Software quality models: A systematic mapping study. In: IEEE/ACM International Conference on Software and System Processes (ICSSP). IEEE (2019), https://doi.org/10.1109/icssp.2019.00025
42. Oriol, M., Marco, J., Franch, X.: Quality models for web services: A systematic mapping. Information and Software Technology 56(10), 1167–1182 (2014), https://doi.org/10.1016/j.infsof.2014.03.012
43. Pedrycz, W., Peters, J., Ramanna, S.: Software quality measurement: concepts and fuzzy neural relational model. In: IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228). IEEE (1998), https://doi.org/10.1109/fuzzy.1998.686259
44. Rawashdeh, A., Matalkah, B.: A new software quality model for evaluating cots components. Journal of Computer Science 2(4), 373–381 (2006)
45. Samoladas, I., Gousios, G., Spinellis, D., Stamelos, I.: The SQO-OSS quality model: Measurement based open source software evaluation. In: IFIP – The International Federation for Information Processing, pp. 237–248. Springer US (2008), https://doi.org/10.1007/978-0-387-09684-1_19
46. Satpathy, M., Harrison, R., Snook, C., Butler, M.: A generic model for assessing process quality. In: New Approaches in Software Measurement, pp. 94–110. Springer Berlin Heidelberg (2001), https://doi.org/10.1007/3-540-44704-0_8
47. Srivastava, S., Kumar, R.: Indirect method to measure software quality using CK-OO suite. In: International Conference on Intelligent Systems and Signal Processing (ISSP). IEEE (2013), https://doi.org/10.1109/issp.2013.6526872
48. Wagner, S., Lochmann, K., Winter, S., Deissenboeck, F., Juergens, E., Hermannsdoerfer, M., Heinemann, L.: The quamoco quality meta-model. techreport TUM-I128, Technische Universität München, Institut für Informatik (2012), https://mediatum.ub.tum.de/doc/1110600/file.pdf

49. Yan, M., Xia, X., Zhang, X., Xu, L., Yang, D.: A systematic mapping study of quality assessment models for software products. In: International Conference on Software Analysis, Testing and Evolution (SATE). IEEE (2017), https://doi.org/10.1109/sate.2017.16
50. Yu, W.D., Radhakrishna, R.B., Pingali, S., Kolluri, V.: Modeling the measurements of QoS requirements in web service systems. SIMULATION 83(1), 75–91 (2007), https://doi.org/10.1177/0037549707079228

All links were last followed on January 17, 2022.