Meta-models for Software Quality and Its Evaluation: A Systematic Literature Review

Nebi Yılmaz[1] [0000-0002-0591-4667] and Ayça Kolukısa Tarhan[2] [0000-0003-1466-9605]

[1,2] Software Engineering Research Group (HUSE),
Hacettepe University Computer Engineering Department, Ankara, Turkey
{yilmaz, atarhan}@cs.hacettepe.edu.tr

Abstract. Software quality has been a very critical issue in software engineering for the last four decades and still maintains its importance. Various quality models with different characteristics have been proposed for quality evaluation of software products. Also, static code analysis tools have been widely used to measure specific characteristics of software in fixed quality models. In order to combine the isolated views on software quality and its evaluation (SQiE), meta-models that formalize whole or partial aspects of software quality models as a base for tool support or further research have been proposed in literature. In this paper, a Systematic Literature Review (SLR) to comprehensively examine the meta-models proposed for SQiE in scientific literature is reported, with an aim to understand their purpose of use, content, and structure. The most-known seven digital libraries were searched, and 28 studies were identified out of 114 initially selected and 6488 initially retrieved in this area between 1997-2020. Results show that majority of meta-models are for general purposes, take ISO 9126 as reference and propose for various types of software. Most of them evaluate quality objectively using metric data and provide quantitative results. Majority of them are structured to enable extension with new quality models.

Keywords: Meta-model, software quality, quality attribute, quality evaluation, software metric.

1 Introduction

It is a challenging task to define the term ‘quality’ for a specific product or service, since its meaning is different for customer, manager, tester, user, developer, etc. This is also true in the field of software engineering, because stakeholders have different expectations from software products or services. There are many definitions of the term ‘software quality’ by different sources. For instance, IEEE [1] defines it as “*the degree to which a system, component, or process meets customer or user needs or expectations*”. The ISO 9001 standard [2], by contrast, defines it as “*the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs*”. The difference in definitions of software quality, in addition to its abstractness and relativity, has made the evaluation of software a challenging task.

Nevertheless, the lack of a common ground to define the term 'software quality' neither decreases its criticality nor the need for its evaluation in the field. In addition, it increases the quality related costs, some of which are catastrophic. Poor quality of software in sensitive systems, such as real-time systems and control systems, may lead to loss of human life, permanent injury, mission failure, dissatisfaction of the users, and increase in the cost of maintenance or financial loss [3]. To eliminate these problems and guide people in evaluating software quality, quality models have been proposed in the field. These quality models are generally composed of a set of quality attributes and relationships between them. The models proposed by Boehm [4] and McCall [5] are the first ones to define quality attributes. These models are followed by international standards such ISO 9126 [6] and ISO 25000 SQuaRE series [7], which propose exhaustive definitions of quality attributes, measurement criteria, and relationship between them. These models define 'software quality' by decomposing it with well-known quality attributes such as reliability, usability, maintainability, etc., which in turn are subdivided into more specific sub-attributes.

Although many quality models have been proposed in the field as outlined above, they are not comprehensive and complete and there are still many issues which pose challenges to their adoption [8]. For example, some models do not cover the entire life cycle of software and some of them do not have a clear vision to explain the correlation between metrics and criteria [9]. Some quality models cover all aspects of software quality but metrics are not consistent with their own conceptual definitions [9]. Also, these quality models are stated to provide either abstract quality attributes or concrete quality assessments. There are no models that seamlessly integrate both aspects [10]. Besides, static code analysis tools are widely used to measure specific characteristics of software quality in fixed quality models. Combining these isolated models and heterogeneous results of code analysis tools to achieve a more complete picture of software quality becomes a main challenge [9][11].

In coping with the ambiguities and problems mentioned, there seems a need to represent the concepts of software quality and evaluation more formally, and meta-modeling can be a suitable vehicle to do this. Researchers have proposed meta-models based on existing quality models, e.g., [10][12]. These meta-models are expected to combine the isolated views to achieve a more complete picture of software quality and in turn, to create a common understanding between stakeholders for proper quality management throughout the entire life of a software product. Accordingly, in order to examine comprehensively the content and structure of the meta-models proposed for software quality and its evaluation (SQiE) in scientific literature, a Systematic Literature Review (SLR) study was carried out and its results are reported in this study. An SLR is a means to evaluate and interpret available research relevant to a particular research hypothesis, topic area, or phenomenon of interest [13]. In this regard, the most-known seven academic search engines (namely Google Scholar, ScienceDirect, Scopus, ACM, Web of Science, IEEE Xplore, and Springer) were used to survey the literature and determine the primary studies for the SLR of the meta-models for SQiE. Only 28 studies out of 114 initially selected and 6488 initially retrieved were identified for further analysis with respect to inclusion and exclusion criteria. These primary studies were analyzed with respect to a number of research questions. To the best of our knowledge, this is the first SLR study conducted on the meta-models for SQiE.

The rest of the paper is organized as follows: Section 2 discusses related work. Research method in conducting this SLR is described in Section 3. Results of SLR are presented in Section 4. Important findings are discussed in Section 5. Threat to validity is discussed in Section 6 and finally, conclusion is stated in Section 7.

2 Related Work

Studies that systematically analyzed models for quality or its assessment are summarized below, since meta-models are created considering those models.

Nistala et al. [14] conducted a systematic mapping of studies that proposed quality models. They examined 238 primary studies and reported that 40 of these studies suggested a new quality model. The authors analyzed these 40 studies in terms of model elements and the support offered towards architecting quality using Bayer's reference architecture framework [15] which proposed an architecture paradigm consisting of planning, realization, documentation, and assessment phases. They concluded that quality attribute and quality metric are the most common model elements in quality models. They also concluded that quality planning and assessment phases are well supported by various quality models, while quality documentation is moderately supported and quality realization is least supported by quality models [14].

Tomar and Thakare [16] conducted a systematic mapping study of quality models, which analyzed 70 relevant primary studies. They determined four research questions and two of them are used to analyze the studies in terms of the most used investigation method and the most used research approach for software quality evaluations. They concluded that most investigated methods are Genetic Algorithm, Neural Network, Tree Decision, Fuzzy logic, Classification, and Regression Tree, while the most used research approaches are Case Study and Experiment Design.

Yan et al. [17] conducted a systematic mapping study of quality assessment models (QAMs) for software products. Their work focused on QAMs from the following aspects: software metrics, quality factors, evaluation methods and tool support. In particular, the authors emphasized the lack of tool support to use quality models in practice throughout the software engineering lifecycle. They concluded that most existing tools are not suitable for industrial requirements.

Cote et al. [18] compared four important quality models (namely ISO 9126, Dromey, Boehm, MacCall) to determine the model that is the backbone of the software engineering field. They used three comparison criteria in their study: five different perspectives of quality, usable from the top to the bottom of the lifecycle, and from the bottom to the top of the lifecycle. The authors concluded that the most suitable model for software engineering is the ISO 9126 quality model. For the same purpose, Al-Qutaish [3] compared five quality models according to the recommended quality features in each model and similarly concluded that the ISO 9126 quality model is the most suitable quality model.

Miguel et al. [19] analyzed 14 quality models in their review article: 6 basics, 4 tailored, and 4 open source. They compared basic and tailored quality models and concluded that the ISO 9126 quality model is the most suitable quality model. They also concluded that ISO 9126/ISO 25010 is the main reference model and that it is needed to incorporate communications as a quality factor in the model [14].

Consequently, by looking at the studies that overview software quality models in the literature, it is observed that some of them systematically investigate the quality models and some of them are limited to comparing only a few quality models. Meanwhile, no systematic study on the studies proposing meta-models for SQiE has been found. Therefore, this study aims to address this gap by a systematic literature review.

3 Research Method

In this study, Systematic Literature Review [13] is used as research methodology in order to analyze studies that proposed meta-models for SQiE. The process performed manually in this study is shown in Figure 1. First of all, research questions which are detailed in Section 3.1 were determined, and the study progressed based on these questions. Then, search strategy was developed and search string was identified to determine publications. The most-known seven academic search engines were used to run the search string, and the initial (1st) set of publications was determined. Exclusion/inclusion criteria were identified and applied while reviewing titles and abstracts of the studies. Then, a refined (2nd) set of publications out of the initial set was identified. Afterwards, exclusion/inclusion criteria were developed and applied while reviewing full texts of the studies, and a limited (3rd) set of publications were determined. Finally, backward and forward snowballing was applied on this set and the research pool was finalized with 28 primary studies

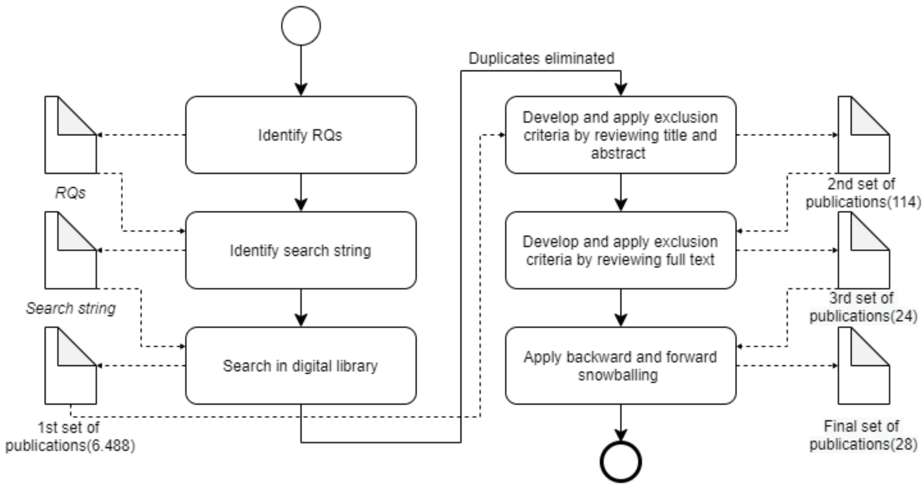

Fig. 1. Process steps used in SLR

Table 1. Research Questions of SLR

RQ#	Description
RQ.1	**What are the basic characteristics of the meta-model proposed in the study?**
RQ.1.1	What is the main purpose of the meta-model proposed? (e.g. generic or specific)
RQ.1.2	Which type of software products are targeted for SQiE? (e.g. OSS, COTS, custom)
RQ.1.3	Is the meta-model taken as the base for tool development in the study? (yes or no)
RQ.2	**Are there any software quality models taken as reference for the proposal? If yes:**
RQ.2.1	Which software quality model(s) are taken as reference? (e.g. ISO 25000)
RQ.2.2	Does the meta-model serve for SQiE with respect to all the models taken as reference?
RQ.2.3	Is the terminology of the software quality model(s) taken as reference mapped to the terminology defined by the meta-model in the study?
RQ.2.4	Is the structure of the software quality model(s) taken as reference mapped to the structure of the meta-model in the study?
RQ.3	**What are the basic characteristics of SQiE as defined in the meta-model?**
RQ.3.1	What methods/techniques are used as reference for SQiE? (e.g. GQM)
RQ.3.2	Does the meta-model support subjective or objective evaluation?
RQ.3.3	Does the meta-model support qualitative or quantitative evaluation?
RQ.3.4	Which data analytics methods are defined for SQiE in the meta-model? (e.g. statistical, machine learning, expert evaluation, fuzzy)
RQ.3.5	How are the results of evaluation provided to users? (e.g. single index, table, graphic)
RQ.3.6	Does the meta-model support SQiE in a specific phase in software development? If yes, which phase is it? (e.g. requirements, coding, field-use)
RQ.3.7	Does the meta-model support SQiE at a single point or throughout software evolution?
RQ.4	**How is the meta-model structured?**
RQ.4.1	Is there a specific structure of the meta-model? If yes, what is it? (e.g. hierarchical)
RQ.4.2	What are the entities defined in the meta-model?
RQ.4.3	Is the meta-model structured to define/include new quality models in evaluation?
RQ.5	**What are the means of data acquisition as defined in the meta-model?** (e.g. manual entry, batch import, automatic transfer from other repositories)
RQ.6	**Has the meta-model been validated? If yes, what was the method of validation?** (e.g. case study, literature mapping, peer review)
RQ.7	**How was the meta-model developed?**
RQ.7.1	Was there a research method employed for development? If yes, what was it?
RQ.7.2	What were the challenges faced in developing the meta-model?

3.1 Research Questions (RQs)

The aim of this SLR was to examine comprehensively the meta-models proposed for SQiE in scientific literature. In order to analyze these studies in detail, the research questions (RQs) listed in the Table 1 were determined. While conducting this study, the PICO template (Population, Intervention, Comparison, Outcomes) which was proposed by [13] is followed:

- Population: Software Quality and Its Evaluation (SQiE)
- Intervention: Meta-models for SQiE

- Comparison: Characteristics of the meta-models proposed
- Outcomes: Purpose and software targeted, entities covered, software quality models or evaluation methods taken as reference, structure and phases employed, data acquisition and analytics methods, validation methods of proposals.

3.2 Search strategy

Before determining the search string to be used in this study, searches were conducted with various combinations of search keywords to obtain the most relevant studies. As a result, the following search string was obtained:

("Meta model" OR "Meta-model") AND ("software quality") AND ("evaluation" OR "assessment" OR "measurement")

We used this search string to retrieve publications in the following digital libraries: Google Scholar, ScienceDirect, Scopus, ACM, Web of Science, IEEE Xplore, and Springer. Among these libraries, Google Scholar was used to increase the coverage of our search. Studies initially obtained from the digital libraries are shown in Table 2. It was observed that the number of studies in Google Scholar was the highest before applying the inclusion/exclusion criteria and Springer followed it. The least number of studies was obtained from Web of Science database.

Table 2. List of digital libraries and number of publications retrieved

Digital Library	URL	Number of publications
Web of Science	apps.webofknowledge.com	7
IEEE Xplore	ieeexplore.ieee.org	19
SCOPUS	www.scopus.com	50
ACM Digital Library	dl.acm.org	152
ScienceDirect	www.sciencedirect.com	216
Springer	www.springer.com	374
Google Scholar	scholar.google.com	5.670
TOTAL		**6.488**

3.3 Publication Selection

By applying the exclusion criteria, many studies from the 1st set of publications were eliminated. SLR guideline [13] was taken as the basis when determining the exclusion criteria. The following exclusion criteria were applied to the 1st set of publications: 1) Duplicate articles; 2) Articles that are not in English; 3) Not formally reviewed articles such as tutorials, sessions, workshop, keynotes, corrigendum and panel; 4) Books and thesis; 5) Articles that do not cover the meta-model for SQiE.

Inclusion criteria were applied to ensure that the publications were within the scope of our study. It was considered that the studies are concerned with SQiE, they propose meta-models accordingly, and they provide mature enough graphical representations of the meta-models including a few entities and relationships between these entities.

In order to apply the inclusion/exclusion criteria, both authors assigned numbers to the articles as '0' or '1', independently, by reviewing title, abstract and keywords. The number '0' meant the study was to be excluded, while the number '1' meant the study was to be included. Articles were included when a total of '2' was obtained, excluded when a total of '0' was obtained, and marked for later discussion when a total of '1' was obtained. As a result, 114 studies were identified, including the conflicting studies (the 2nd set of publications). After a series of discussions, conflicts were resolved.

Both authors reviewed the full texts of the studies in the 2nd set of publications, applying the inclusion and exclusion criteria, and 24 studies were selected (as the 3rd set of publications). Then, backward and forward snowballing [20] were applied to improve the coverage and 4 more articles were included in the research pool. As a result, a total of 28 publications listed in [21] were selected for detailed analysis.

3.4 Data Extraction

Before starting data extraction, the initial categories for each RQ shown in Table 1 were determined by the authors. When there was a need to add a new category during data extraction, this was discussed by the authors and added as a category item in data extraction sheet. The first author answered the RQs by reading the full texts of the studies, and recorded elicited information by the categories in a tabular form. Then, peer-review was performed by the second author and any conflict between the authors was resolved in a series of discussions. Data extraction sheet can be reached by the following link: https://tinyurl.com/ybz2ybky

4 Results

4.1 Basic characteristics of meta-models (RQ 1)

RQ 1.1: A meta model is defined as "a model of a well-defined language" [22] or "a model of the models" [23]. As shown in Fig. 2 (a), 75% of the meta models were proposed for general purposes. These introduce the fundamental concepts present in every single approach to fixed quality models. They are abstract enough to be used in several software engineering activities: specification, design, development, certification, selection, etc [24]. Fig. 2 (a) also shows that 25% of the meta models were proposed for specific purpose. These are either developed for a specific software type (e.g. web services) or proposed for a specific phase in software development process.

RQ 1.2: As shown in Fig. 2 (b), more than half of the meta-models were proposed to cover all types of software. Only 2 of them were proposed for open source software (OSS), 2 of them for commercial software (COTS), and 1 of them for microservices (MS) software. In addition, 5 meta-models were proposed for web services (WS).

RQ 1.3: Time is a crucial factor in reducing software evaluation costs. Tools that allow automatic evaluation and eliminate manual effort have critical importance during software quality evaluation. As shown in Fig. 2 (c), less than half of the studies developed tools to reduce the effort spent on software quality evaluation.

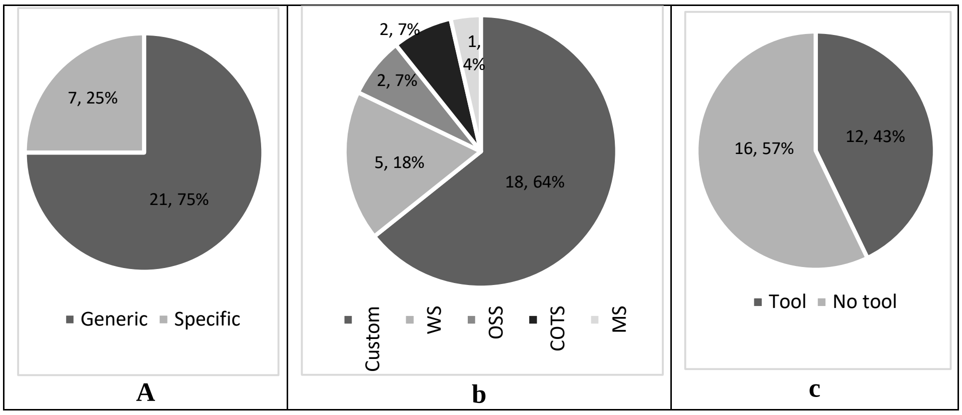

Fig. 2. Basic characteristics of meta-models (RQ1): **(a)** Percent distribution of main purpose, **(b)** Percent distribution of types of software products targeted, **(c)** Percent distribution of whether meta-models are taken as the base for tool development.

4.2 Software quality models referenced in developing meta-models (RQ 2)

RQ 2.1: Existing quality models were taken as reference in creating most of the meta-models. As shown in Fig. 3 (a), more than half of the proposed meta-models took ISO 9126 as reference. Similarly, 6 of them took ISO 25010, 8 of them took McCall's model, 9 of them took Boehm's model, 6 of them took Dromey's model, 3 of them took IEEE 10610 as reference, and 6 of them took other existing meta-models as reference. In 3 studies, quality models referenced are not explicitly specified. It should also be noted that some meta-models took one or more quality models as reference. Earlier meta-models were based on McCall and Boehm models, while later meta-models were based on ISO 9126. All referenced quality models are hierarchical in structure. Therefore, most of the meta-models examined the hierarchical structure of the quality models, and their objective was to capture the knowledge on software quality present in the hierarchical quality models in one comprehensive model.

RQ 2.2: As shown Fig.3 (b), 11 of meta-models serve for SQiE with respect to all models taken as reference while 6 of them do not. The rest of them does not explicitly specify if they serve for SQiE with respect to all quality models taken as reference.

RQ 2.3: As shown in Fig. 3 (b), while 11 studies map the terminology of software quality models taken as reference to the terminology of the meta-models they propose, 5 studies do not perform this mapping. Also, 7 studies do not explicitly specify whether they mapped the terminology of the quality models they referenced, and 5 studies make this mapping only partially.

RQ 2.4: As shown in Fig. 3 (b), 10 studies map the structure of the software quality model taken as reference to the structure of the meta-models they propose, and 6 studies do not perform this mapping. In addition, 10 studies do not explicitly specify whether they map the structure of the quality models they referenced, and 3 studies make this mapping only partially.

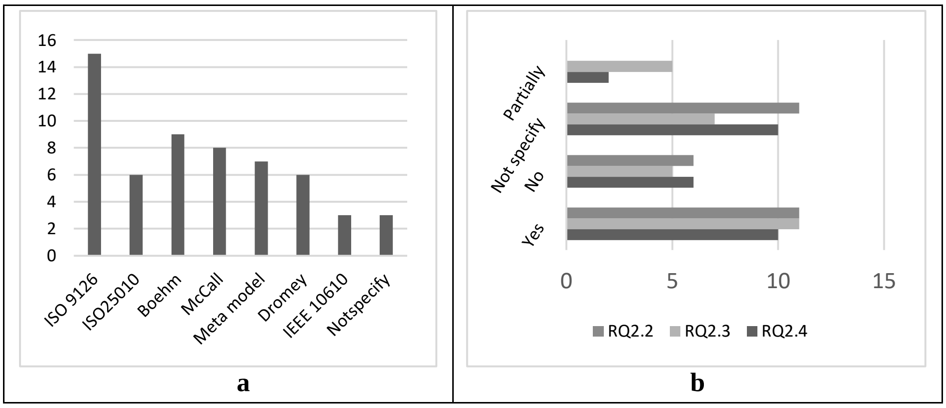

Fig. 3. **(a)** Software quality model(s) taken as reference for meta-models (RQ2.1), **(b)** Number of studies for RQ2.2, RQ2.3 and RQ2.4.

4.3 Basic characteristics of SQiE as defined in meta-models (RQ 3)

RQ 3.1: We observed 11 meta-models use Goal-Question-Metric (GQM) method and 3 meta-models use Factor-Criteria-Metric (FCM) method for quality evaluation. The remaining (50%) studies do not explicitly specify the methods they use.

RQ 3.2: As shown in Fig. 4 (a), 14 meta-models evaluate quality objectively using only metric data. Only 1 study does not explicitly specify whether it uses metric data or user opinion in evaluation. The rest (46%) of studies make both objective and subjective evaluations considering both metric data and user opinions. In almost all studies, meta-models use metric data in quality evaluation. To analyze the metrics, they either develop their own tools or import data using the existing tools.

RQ 3.3: We observe from Fig. 4 (b) that only 1 study gives qualitative result after evaluation, and 2 studies do not explicitly specify whether they provide quantitative or qualitative results. Also, 7 studies give both quantitative and qualitative results after evaluation. The rest of the studies (63%) provide quantitative results only. These findings show that meta-models are generally aimed to produce quantitative values in order to see concrete results after evaluating software quality.

RQ 3.4: As shown Fig. 4 (c), 3 studies use expert decision, 1 study uses survey, 2 studies use fuzzy logic, 1 study uses analytic hierarchy process (AHP), 2 studies use statistical methods, 2 studies use human decision, and 1 study uses machine learning (ML) to analyze data in their meta-models. The remaining studies (42.8%) do not use any of these data analytics techniques.

RQ 3.5: As shown in Fig. 4 (d), 9 studies provide evaluation results as an index in range [0,1] and 2 studies provide evaluation results as an index in range [0, max]. These results enable the comparison between the alternative products. Also, 3 studies provide evaluation results in Likert scale, and 4 studies provide results in graphical representation. However, 4 studies do not explicitly specify how the evaluation results are provided. Overall, many studies (64.2%) provide numerical values. It should be noted that one study might have one or more of the result types mentioned above.

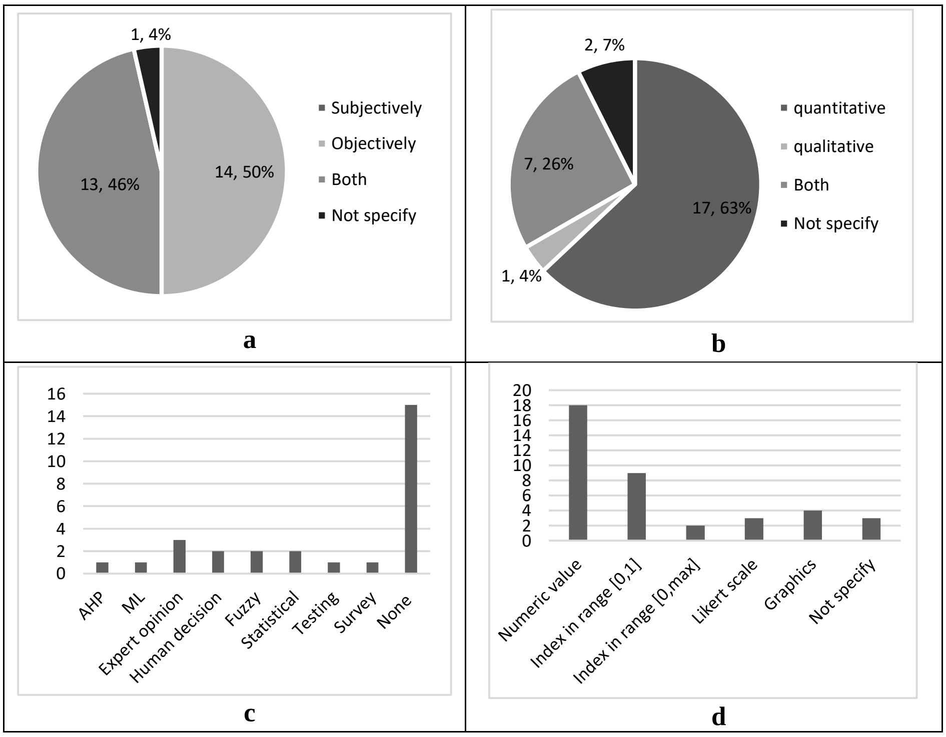

Fig. 4. Basic characteristics of SQiE in meta-models (RQ3): **(a)** Percent distribution of subjective/objective evaluation, **(b)** Percent distribution of quantitative/qualitative evaluation, **(c)** Numeric distribution of analytics methods, **(d)** Numeric distribution of evaluation result types

RQ 3.6: We observed that 12 studies support SQiE in software delivery, and 10 studies do not support it in a specific phase. Only 1 study supports SQiE in implementation and 1 study in requirements; 4 studies do not explicitly specify a phase. Since product delivery is important for end users, it is the most addressed phase.

RQ 3.7: We observed that only 5 studies support SQiE in the evolution of software while the rest supports it at a single point. There are many changing factors in software lifecycle. Since changes are inevitable, the software needs to keep up with those factors. However, majority of meta-models support SQiE at a single point in time.

4.4 Structure of meta-models (RQ 4)

RQ 4.1: Most quality models have special structures such as hierarchical. Like quality models, meta-models also can have special structures. For instance, the meta-model in study [10] consists of 2-layers for specification and evaluation. In another study [24], the meta-model consists of 3-layers for fundamental concept, metric, and context. In this SLR, it was observed that 12 studies propose meta-models having a layered structure and that the remaining (57%) studies do not have any specific structure.

Table 3. Terms used in meta-models as entities with their frequencies

Category	**Names of entities in different meta-models**	**Freq.**
Data analysis	**Syn:** Analysis model / decision criteria / interpretation rule / analysis	7
Entity	**Syn:** Entity / component / quality artifact / measurable entity	18
Evaluation (E)	**Syn:** Evaluation / assessment model **Agg:** Formula / rule / E. result / E. aspect / E. impact	14
Instrument	**Syn:** Tool / instrument	5
Measure	**Syn:** Measure / metric **Agg:** Base measure / base metric / derived measure / derived metric	23
Measurement (M)	**Agg:** M. approach / measurable concept / M. method / M. function / M. data / M. result / value / indicator	13
Property	**Syn:** Property / quality aspect / quality dimension / quality type / feature	12
Quality attribute	**Syn:** Quality characteristic / quality attribute / quality factor / characteristic / attribute / factor / product factor **Agg:** Sub-characteristic / sub-factor / base attribute / direct attribute / derived attribute / indirect attribute	27
Quality goal	**Syn:** Quality goal / goal / quality target / purpose / target / objective	7
Quality model	**Syn:** Quality model	9
Requirement	**Syn:** Quality requirement / requirement / specification	3
Scale	**Syn:** Scale / type of scale / measurement scale	4
Unit	**Syn:** Unit / measurement unit / unit of measurement	7
View	**Syn:** Viewpoint / view / stakeholder	6

* **Syn** denotes synonymous concepts for a category, while **Agg** denotes sub-categories or aggregated concepts under a category.

RQ 4.2: Meta-models are diagrams that contain entities and relationships between entities. Entities for SQiE were sometimes defined by different terms in meta-models, even though they were defined for the same concept. Therefore, there seems a confusion in literature in the terminology proposed for SQiE. The terms used for concepts represented as entities in meta-models were analyzed in detail and assigned to category groups shown in Table 3. Accordingly, 'quality attribute' and 'measure' are the most commonly used entities, while 'requirement', 'scale' and 'instrument' are the least frequently addressed entities in the meta-models.

RQ 4.3 Meta-models for SQiE are expected to cover all aspects of quality models and be flexible enough to apply with modifications. They should be organized according to the needs of users and enable inclusion of new models. Majority (24) of the meta-models were structured to enable extension with new quality models when needed. Also, we observed that only 2 studies do not allow to define new models and 2 other studies do not explicitly specify if they enable to define new quality models.

4.5 Data acquisition addressed in meta-models (RQ 5)

Some studies developed their own tools based on the meta-models as addressed in response to RQ 1.3. In addition, the majority of the meta-models use metric data for evaluation as specified in response to RQ 3.2, and obtain quantitative values as a result as addressed in RQ 3.3. In order to obtain quantitative values, the tools based on the meta-models are related to some external analysis tools. Among the meta-models examined in this SLR, 6 use batch import to acquire data from external analysis tools, while 6 analyze data after automatic direct transfer from other repositories. Also, 7 meta-models allow manual data entry. Some of the meta-models support one or more data acquisition methods mentioned above. However, many studies (15) do not explicitly specify how their meta-models acquire data.

4.6 Validation method of meta-models (RQ 6)

Validation of the proposed meta-models is one of the most important steps in the studies, since research validation is vital to ensure the research is clean, correct and useful. According to the findings obtained, 13 meta-models were validated by designing case studies and 6 meta-models by performing toy experiment. Also, 4 studies conducted peer reviews by experts and 1 study used pilot project application to validate its meta-model. While 5 studies did not explicitly mention the method of validation for their proposals, only 1 study did not use a validation method. A study might have been validated with peer reviews along with a case study or toy experiment.

4.7 Development of meta-models (RQ 7)

RQ 7.1: Research methods employed in proposing the meta-models were classified according to the scheme by Wieringa [25]: solution proposal, weak empirical study, strong empirical study, opinion paper, experience paper, and philosophical paper. While 13 meta-models were classified as solution proposals with either weak demonstration or hypothetical example, 10 meta-models developed under a lab experiment were classified as weak empirical study. Remaining 5 studies, which implemented research in practice by setting hypotheses, were classified as strong empirical study.

RQ 7.2: In the studies examined, it is mentioned that many challenges were faced while creating the meta-models. These challenges, which arose especially from the quality models taken as reference, were classified based on the suggestions of [26]. In Table 4, the challenges faced in proposing the meta-models and the number of studies (with percent distribution) facing these challenges are given. Accordingly, the poor interpretation of interdependencies and measurements was one of the most common challenge. Inconsistency among different terminologies was another important challenge. Only 4 studies reported the challenge of different expectations of stakeholders.

Table 4. Classification of challenges and the number of studies having faced these challenges

Description of challenge	#studies
C1: Inconsistency in terminology: Most approaches that are not based on theoretical grounds, lack a definition for quality concepts that is precise and concise [26].	**9** (24%)
C2: Partially defined: Most quality models are outlined but not fully developed. All define measurable concepts, some of them also attributes, few of them include (most often partial) measures and scarcely any defines decision criteria or indicators [26].	**7** (18%)
C3: Lack of focus: Most quality models provide an extensive (and mostly tangled) coverage of stakeholders and levels of abstraction [26].	**7** (18%)
C4: Lack of clarity in interdependencies and measure interpretations: In most quality models that are not based on theory, the degree of influence of individual internal quality factors on the quality in use of the application, as well as their interdependencies, are not well established [26]. Also, measure interpretations of some models are not clear.	**11** (30%)
C5: Different expectations of stakeholders: Stakeholders in the software process has different expectations from meta-models	**4** (10%)

5 Summary of Findings and Discussion

Regarding the basic characteristics (RQ1), 75% of meta-models were proposed for generic purposes. This situation coincides with the purpose of proposing meta-models in that they should have a general feature shaped according to the purpose of the user. At the same time, the vast majority of meta-models were proposed to evaluate custom software products. However, when the empirical studies for testing the validity of meta-models are examined, it is observed that almost no meta-model used the specific metrics of OSS such as mailing list, user reviews, etc. Therefore, it is not clear whether these meta-models are fully applicable to evaluate the quality of OSS products.

One of the most important contribution of the studies is tool support. Since code analysis tools measure specific features of software quality in fixed quality models, it might be necessary to combine the heterogeneous results of code analysis tools to measure quality more comprehensively and completely. In such a case, inconsistent results can be obtained. To eliminate this, tool support for meta-models is essential. However, more than half of the proposed meta-models did not provide tool support.

Regarding the quality models taken as reference while developing the meta-models (RQ2), ISO 9126 was the leading model in 57% of the studies since: 1) it provides a comprehensive specification and evaluation model for software product quality, 2) it explicitly addresses user needs of a product by allowing a common language for specifying user requirements by various stakeholders, and 3) it evaluates quality of software products objectively based on observation and not opinion [27]. However, this quality model was withdrawn and replaced by ISO/IEC 25010 [7].

Regarding the basic characteristic of SQiE (RQ3), Goal-Question-Metric method was used for quality evaluation in the meta-models. Also, metric data was very important in evaluating software, since majority of the meta-models support objective and quantitative evaluations and obtain numerical values as the result of their evaluations. However, there are very few meta-models that used the well accepted data analytics methods such as ML, Fuzzy, AHP etc. Generally, meta-models analyzed data independent of these important methods.

There are many factors that change throughout the software lifecycle. Therefore, changes to software are inevitable, as the software needs to keep up with these changing factors. However, most of the meta-models (82%) did not consider the evolution of the software product and only focused on the last version when evaluating quality. This may be desirable for end users, but it creates a difficulty for developers in monitoring the quality while maintaining the software product. Considering that 60% of the total cost in software projects is spent for product maintenance [28], the importance of this issue is better understood in meta-modeling of SQiE.

Regarding the structure of the meta-models (RQ4), 43% of them are layered with a special structure, and the rest does not have a special structure. Majority (86%) of the meta-models are structured to enable extension with new quality models, which is in line with the purpose of meta-modeling. In addition, regarding the content of the meta-models, ‘quality attribute’ and ‘measure’ are the most commonly used entities.

Regarding data acquisition (RQ5), while almost half (13) of the meta-models adopt batch import (6), automatic direct transfer from other repositories (6), and manual entry (7), the rest of them (15) do not explicitly specify how they acquire data. Please note that a meta-model may support more than one data acquisition method.

With respect to the empirical evidence (RQ6), meta-models were validated by either hypothetical case studies or toy experiments. A real-world case was not used for validation of any meta-model. Therefore, it is observed that there is no empirical evidence sound enough to demonstrate usefulness of the meta-models for SQiE.

Regarding the research methods employed (RQ7) in developing the meta-models, almost half of the studies (47%) are classified as solution proposal. It means that meta-models were generally proposed as solutions for SQiE and argued for its relevance, without a full-blown validation. Therefore, they offered either a weak, hypothetical example or demonstration. Also, regarding the challenges faced in creating the meta-models, it was observed that majority of the challenges arose especially from the quality models taken as reference. Because most quality models are not based on theory, the degree of influence of individual internal quality factors on the quality in use of the application as well as their interdependencies are not well established [26]. Also, some quality models cover all aspects of software quality but metrics are not consistent with their own conceptual definitions [9].

6 Threats to Validity

The purpose of this SLR was to focus on and analyze specific studies in literature. This situation may lead to problems related to the acquisition of all relevant studies and hence the validity of the results. Main threats might have been due to the choice of search string as well as biases in study selection, data extraction, and classification.

In order to select the search string, first we piloted the searches and reviewed the results in multiple iterations; then, we updated search keywords after the reviews. To alleviate the threat of the search string, backward and forward snowballing were applied to obtain potentially relevant publications that were missed by the search string.

There might have been researcher bias in identifying the exclusion/inclusion criteria for choosing the studies. To alleviate the threat in study selection, a systematic voting process was carried out by the authors independently to decide which articles should be included in the pool. In this way, it was aimed to minimize subjectivity and bias in selecting the publications. The studies that the authors agree on were included in the pool. Other studies were marked for later discussion, and conflicts were solved between the authors after a series of discussions. Non-agreed studies were excluded.

To cope with the threat of data extraction, the first author answered the research questions by reading the full texts of the studies in the final pool, and recorded elicited information by the categories in a tabular form. Then, peer-review was performed by the second author in detail and if there was a conflict between authors, it was resolved after a series of discussions among the authors.

Some studies were used as reference to alleviate inconsistency in the schemes of classification. For example, research methods used in developing the meta-models were examined in RQ 7.1. While answering this research question, studies were classified according to the classification scheme by Wieringa [25]. Also, the challenges faced while developing the meta-models were investigated in RQ 7.2. To answer this question, the study [26] for classifying the difficulties was taken as reference.

In addition, during SLR process depicted in Figure 1, search string, search process and selection criteria were piloted first and refined in a series of iterations. Whenever an update was required after an iteration, the whole process was repeated again.

7 Conclusion

In this paper, the results of an SLR study are reported in order to examine comprehensively the content and structure of the meta-models proposed for SQiE in scientific literature. The most-known seven digital libraries were searched, and 28 studies out of 114 initially selected and 6488 initially retrieved were identified for further analysis with respect to the inclusion/exclusion criteria. These primary studies were analyzed with respect to seven research questions. To the best of our knowledge, this is the first SLR study conducted on the meta-models for SQiE.

This SLR study might help researchers and practitioners in understanding the state of the art on the meta-models proposed for SQiE. The set of 28 studies in the pool serves as a reference catalogue for researchers, and the detailed analysis provides a kind of guide in seeing the weak aspects to propose further meta-models or design further studies. The studies that propose tools based on the meta-models, on the other hand, might be useful for practitioners for investigation and adoption for their use.

References

1. IEEE Std. 610.12: Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, New York, NY, USA, 1990.
2. ISO. ISO/IEC 14598-1: Software product evaluation - Part 1: General overview. International Organization for Standardization, Geneva, Switzerland, 1999.
3. Al-Qutaish, R. E.: Quality models in software engineering literature: an analytical and comparative study. In: Journal of American Science. 6(3): 166-75, 2010.
4. Boehm, B. W., Brown, J. R., Lipow, M.: Quantitative evaluation of software quality. In: Proceedings of the 2nd Int Conf on Software Eng, Oct 13, pp. 592-605, IEEE, 1976.
5. McCall, J.A., Richards, P. K., Walters, G.F.: Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality, Fort Belvoir, VA: Defense Tech Info Center, 1977.
6. ISO/IEC TR 9126, Software engineering–Product quality – Part 1,2,3. 2002-03-15.
7. ISO 25000: Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE) series of standards–Quality Management Division, 2014.
8. Thapar, S. S, Singh, P., Rani, S.: Challenges to development of standard software quality model. International Journal of Computer Applications. 49(10), 2012.
9. Khammal, A., Boukouchi, Y., Hanine, M., Marzak, A.: General Meta-model of Software Quality. In: (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (4), pp. 1728-1734, 2016.
10. Wagner, S., Lochmann, K., Heinemann, L., Kläs, M., Trendowicz, A., Plösch, R., Seidi, A., Goeb, A., Streit, J.: The quamoco product quality modelling and assessment approach. In: 34th International Conference on Software Engineering, pp. 1133-1142, IEEE, 2012.

11. Deissenboeck, F., Heinemann, L., Herrmannsdoerfer, M., Lochmann, K., Wagner, S.: The quamoco tool chain for quality modeling and assessment. In: 33rd International Conference on Software Engineering (ICSE), May 21, pp. 1007-1009, IEEE, 2011.
12. Boukouchi, Y., Khamal, A., Marzak, A., Moutachaouik, H. A.: MetaModel for Quality Software Based on the MDA Approach. In: International Journal of Computer Science and Information Technologies, Vol. 5 (3), pp. 4390-4393, 2014.
13. Keele,S.: Guidelines for performing Systematic Literature Reviews in Software Engineering, Version 2.3, EBSE-2007-01.
14. Nistala, P., Nori, K. V., Reddy, R.: Software quality models: A systematic mapping study. In: IEEE/ACM Int Conf on Software and System Processes (ICSSP), pp. 125-134, May 25, IEEE, 2019.
15. Bayer, J., Forster, T., Ganesan, D., Girard, J.F., John, I., Knodel, J., Kolb, R., Muthig, D.: Definition of Reference Architectures based on Existing Systems, p. 173, 2004.
16. Tomar, A. B., Thakare, V. M.: A Systematic Study of Software Quality Models, In: International Journal of Software Engineering & Applications, vol. 2, no.4, pp.61–70, 2011.
17. 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), pp. 63-71, IEEE, Nov 3, 2017.
18. Côté, M.A., Suryn, W., Georgiadou, E.: In search for a widely applicable and accepted software quality model for software quality engineering, In: Software Quality Journal, vol. 15, no. 4, pp. 401–416, Nov 2007.
19. Miguel, J. P., Mauricio, D., Rodríguez, G.: A Review of Software Quality Models for the Evaluation of Software Products, In: International Journal of Software Engineering & Applications, vol. 5, no. 6, pp. 31– 53, Nov 2014.
20. Webster J., Watson R. T.: Analyzing the past to prepare for the future: writing a literature review. MIS Q 26: xiii—xxiii, 2002.
21. Yilmaz, N., Tarhan, A.K.: Study Pool for Meta-models for Software Quality & Its Evaluation: A Sys Lit. Review [online], Available: https://zenodo.org/record/3948238#.XxB3BSgzaUm
22. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture Practice and Promise. Addison Wesley, 2003.
23. OMG: Model Driven Architecture (MDA). Object Management Group. OMG document ormsc/2001-07-01.
24. Burgués, X., Franch, X., Ribó, J.M.: A MOF-compliant approach to software quality modeling. In: International Conference on Conceptual Modeling, Oct 24, pp. 176-191, Springer, Berlin, Heidelberg, 2005.
25. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements Engineering, vol. 11, no. 1, pp. 102–107, Mar 2006.
26. Cachero, C., Calero, C., Poels, G.: Metamodeling the quality of the web development process' intermediate artifacts, In: International Conference on Web Engineering, pp. 74-89, Springer, Berlin, Heidelberg, Jul 16, 2007.
27. Al-Kilidar, H., Cox, K., Kitchenham, B.: The use and usefulness of the ISO/IEC 9126 quality standard, In: International Symp on Empirical Software Eng, p.7, IEEE, Nov 17, 2005.
28. Doneva, R., Gaftandzhieva, S., Doneva, Z., Staevsky, N.: Software Quality Assessment Tool Based on Meta-Models, In: International Journal of Computer Science and Mobile Computing, pp. 574-590, 2015.