SN Computer Science

Software Product Quality Models, Developments, Trends and Evaluation

--Manuscript Draft--

Manuscript Number:	SNCS-D-19-00561R2
Full Title:	Software Product Quality Models, Developments, Trends and Evaluation
Article Type:	Original Research
Funding Information:	
Abstract:	Software product quality models have evolved in their abilities to capture and describe the abstract notion of software quality since the 1970's. Many models constructed deal with a specific part of software quality only which makes them ineligible to assess the quality of software products as a whole. Former publications failed to thoroughly examine and list all the available models which attempt to describe each known property of software product quality. This paper discovers such complete software product quality models published since 2000; moreover, it endeavours to measure the relevance of each model quantitatively by introducing indicators with regard to the scientific and industrial community. The identified 23 software product quality model classes differ significantly in terms of publication intensity, publication range, quality score average, relevance score and the 12-month average of the Google Relative Search Index. The results offer a foundation for selecting the appropriate software product quality model for use or for extension if newly identified quality properties need to be connected to a general context. Furthermore, the experiences accumulated on the field of software product quality modelling motivated researchers to successfully transfer the concepts to other areas where abstract entities need to be compared or assessed including the quality of higher educational teaching and business processes, which is also briefly highlighted in the paper.
Corresponding Author:	Tamas Galli, M.Phil., B.Eng., B.Eng. De Montfort University Leicester, Leicestershire UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	De Montfort University
Corresponding Author's Secondary Institution:	
First Author:	Tamas Galli
First Author Secondary Information:	
Order of Authors:	Tamas Galli
	Francisco Chiclana
	Francois Siewe
Order of Authors Secondary Information:	
Author Comments:	Dear Dr. Ciancaglini, Thank you very much for your response and the decision. Please find below our response to the reviewer's comments: Reviewer #1: The authors put a significant effort into improving the paper. I can accept most of the answers and improvements. One thing remains, which might stem from the lack of clarity of my previous review: I asked for deeper insights regarding the collected works that is true, but I also asked for a more concise and compact paper (with less redundances). The authors removed the Discussion but introduced 33 (!) new pages with "deeper insights" about the models. Actually, this is no what I meant to suggest. The 33 new pages contain textual summary of each of the model papers. I am sorry if I

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

	mislead the authors, but what I wanted to see were something like tables with comparison of what each model does/uses, like are the models theoretical or are there any tools implementing it, do they use some kind of benchmarking for comparison, what type of metrics do they rely on, etc. These would take couple of tables, charts, not 33 pages of new text. Nonetheless, I found the terminology for each model to be useful, so keeping some information but in a much much more compacted version is encouraged. To emphasize the above, I suggest the authors to explicitly state which models are "theoretical" and which ones are "practical". Since ISO/IEV 25010 for example a standard that contain guidelines and terminology for software quality measurement, however, it is not practically applicable as is. One needs to adopt and extend it to be useful. The same is true for Droomey, Boehm's, McCall, etc. In contrast, SQALE, SQUALE or QUAMOCO are practical models one can take and apply to get actual numbers for quality measures. This is a very significant difference for example from industrial application point of view. To summarize, I would suggest to move the lengthy comparison text into the appendix, keep only a short terminology description and provide some compariosn tables that compare the various models from different aspects. Authors' response: Thank you for the comments. We tried to comply with your request of moving "the lengthy comparison text into the appendix, keep only a short terminology description and provide some comparison tables that compare the various models from different aspects." However, we thought that the coherent unit and in-depth investigation that we believe these 33 pages have on their own would be detrimental by this moving. Indeed, in our opinion, other publications mentioned in our 'Related Works' section are not as comprehensive as our comparison, where each model is presented in a unified manner including its terminology and concepts. Therefore, we decided to remove it from the paper and instead we added, in the 'Results' section, 2 further tables (table 4 and table 5) with brief explanations to highlight tool support and to indicate whether the assessment of execution tracing or logging quality is supported by the model. The latter is an important aspect in our research. These changes have shorten the paper to 21 pages plus appendix. Thank you for taking the time to read the long paper. Kind regards, Tamas Galli
Response to Reviewers:	Authors' response: Thank you for the comments. We tried to comply with your request of moving "the lengthy comparison text into the appendix, keep only a short terminology description and provide some comparison tables that compare the various models from different aspects." However, we thought that the coherent unit and in-depth investigation that we believe these 33 pages have on their own would be detrimental by this moving. Indeed, in our opinion, other publications mentioned in our 'Related Works' section are not as comprehensive as our comparison, where each model is presented in a unified manner including its terminology and concepts. Therefore, we decided to remove it from the paper and instead we added, in the 'Results' section, 2 further tables (table 4 and table 5) with brief explanations to highlight tool support and to indicate whether the assessment of execution tracing or logging quality is supported by the model. The latter is an important aspect in our research. These changes have shorten the paper to 21 pages plus appendix. Thank you for taking the time to read the long paper.

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Click here to access/download;Manuscript;software_product_quality_models

Click here to view linked References

SN Computer Science manuscript No.
(will be inserted by the editor)

Software Product Quality Models, Developments, Trends and Evaluation

Tamas Galli * · **Francisco Chiclana** · **Francois Siewe**

Received: date / Accepted: date

Abstract Software product quality models have evolved in their abilities to capture and describe the abstract notion of software quality since the 1970's. Many models constructed deal with a specific part of software quality only which makes them ineligible to assess the quality of software products as a whole. Former publications failed to thoroughly examine and list all the available models which attempt to describe each known property of software product quality. This paper discovers such complete software product quality models published since 2000; moreover, it endeavours to measure the relevance of each model quantitatively by introducing indicators with regard to the scientific and industrial community. The identified 23 software product quality model classes differ significantly in terms of publication intensity, publication range, quality score average, relevance score and the 12-month average of the Google Relative Search Index. The results offer a foundation for selecting the appropriate software product quality model for use or for extension if newly identified quality properties need to be connected to a general context. Furthermore, the experiences accumulated on the field of software product quality modelling motivated researchers to successfully transfer the concepts to other areas where abstract entities

Tamas Galli
Institute of Artificial Intelligence (IAI), Faculty of Computing, Engineering and Media, De Montfort University, Leicester, UK
E-mail: tamas.galli@my365.dmu.ac.uk
E-mail: tamas.galli@bcs.org

Francisco Chiclana
Institute of Artificial Intelligence (IAI), Faculty of Computing, Engineering and Media, De Montfort University, Leicester, UK
Andalusian Research Institute on Data Science and Computational Intelligence (DaSCI), University of Granada, Spain
E-mail: chiclana@dmu.ac.uk

Francois Siewe
Software Technology Research Laboratory (STRL), Faculty of Computing, Engineering and Media, De Montfort University, Leicester, UK
E-mail: FSiewe@dmu.ac.uk

need to be compared or assessed including the quality of higher educational teaching and business processes, which is also briefly highlighted in the paper.

Keywords software engineering · software product quality model · quality assessment · execution tracing

1 Introduction

Software product quality models assess the quality properties of software products in contrast to process quality models that evaluate the processes through which the software products come into existence [72, 75]. Modelling, assessing and managing the quality of software products goes back to the 1970's when the first complete software product quality model was defined by Boehm, Brown, and Lipow [18] in 1976, which endeavoured to consider each known aspect of software product quality, followed by McCall, Richards and Walters's quality model in 1977 [114]. Many new software product quality models have been constructed since the 1970's as presented in the next sections. The appearance of the ISO/IEC 9126 standard family [72] in 1991, revised in 2001, urged this process by laying emphases on tailoring the quality model of the standard to the given project's needs.

The quality models born offer considerably different benefits to the software industry and research community when the quality of concrete software products needs to be assessed, different products compared or the development of the same product needs to be investigated at different points in time to define measures. In such contexts, partial quality models that aim to assess one or more but not all known characteristics of the product quality are not eligible. Nevertheless, significant amount of the models created since the 1970's were able to deal with a specific part of software product quality only. In this publication we focus on quality models that aim to define quality properties to cover the whole set of software product quality. We emphasize the aim of such models throughout this paper and designate them as complete software product quality models even if completeness means an elusive target in a precise sense. Furthermore, we rule out the partial software product quality models from our investigation, which give up the aim of the above completeness by purpose.

Conquering the barriers to describe the abstract notion of software product quality in a tangible manner motivated researchers to apply the same concepts and manifested experiences in different fields including higher educational teaching quality and service quality, through which the importance of the topic exceeds the area of software engineering. Information is provided on these quality models in section 3.1.

In addition, we also highlight with the presentation of the different models how they evolved in their abilities to capture software quality. Deisenboeck at al. classify the models according to the activities they support [28]: (1) quality definition, (2) quality assessment and (3) quality prediction. The three areas would theoretically form subsets of each other but the practice shows sharp differences [28]. Models based on the goal-criteria-metric approach [134] usually address only a subset of product quality without the view of the whole; moreover, predictive models show the same restrictions extended by a very limited context of use and the lack of ability to be generalised [28].

Some models offer easy to automate features including SQALE [99] with widespread implementations [115, 128, 135, 139, 158]. However, fully automated model implementations are unable to measure external quality as they consider source code artefacts which undergo static code analysis.

Further important aspect of the topic is delineated by execution tracing quality. Logging, execution tracing often used as synonyms play a key role in identifying software errors in multi-threaded, distributed or embedded application and have a denoting impact through that on the quality property maintainability [19, 45, 146]. Nevertheless, software product quality models offer serious improvement potential with regard to execution tracing quality as pointed out in [41,42]. The current research also contributes to this field by the thorough analysis of quality models introduced.

Previous publications attempting to review software product quality models or parts of them are presented in the section Related Works. However, (1) the previous research conducted was not implemented as a systematic literature review with defined rigorous rules [83], (2) none of these publications measure the relevance of the presented quality models with regard to the scientific and industrial community (3) the publications do not show the latest developments since 2014. The demands and the mentioned shortcomings on the field made necessary to conduct a systematic literature review.

In conclusion, the publication makes a novel contribution (1) by the identification of the software product quality models, which aim at completeness, based on a systematic literature review [83] to minimise and possibly avoid bias, (2) by the definition and introduction of the indicators: relevance score, quality score, quality score average and publication range to show the publication intensity of the quality models identified and through that the associated academic or industrial research interest, (3) by the consideration of the 12-month average Google Relative Search Index [47] to infer how widespread the everyday use cases of the quality models are, and (4) by highlighting the trends and developments on the field. The mechanism how to measure the relevance of the quality models identified constructs an extension from the point of view of research methodology to any systematic literature review.

In the section Research Methodology we outline the research, describe how we measured the relevance of the quality models and how this method can be applied in general in the systematic literature review process. The quality models identified are listed with their relevance scores, quality score averages, publication ranges, and the 12-month average Google Relative Search Indexes [47] in the section Results. The individual quality scores of the publications are listed in the Appendix D. In addition, we devoted a separate section to the examination of reliability and validity of the research in Threats to Validity. Finally, we close the publication with Related Works and Conclusions. For reproducibility, we also publish the query strings performed in the different scientific journal databases, moreover, the result lists delivered by these queries. This supplementary material is placed in the Appendix.

2 Research Methodology

Systematic literature review includes the following major steps: (1) identifying the problem, (2) developing a protocol for the research, (3) defining the research questions, (4) developing the keywords for the search, (5) defining the search strategy and identifying the document databases, (6) defining the inclusion and exclusion criteria, (7) defining the data to be extracted from the documents identified, (8) establishing the evaluation criteria for the documents, (9) searching the document databases, (10) recording the data, and (11) synthesizing, reporting the results [82, 83]. In addition to the above steps, each publication included in the research is scored individually. This individual score value we name quality score and it encompasses how clear and how actual the given publication is as explained in section 2.5.

The identified publications introduce, tailor or adapt software product quality models. Thus, each publication can be assigned to a software product quality model. This way we build clusters of related publications labelled with the given software product quality model and call them software product quality model classes or software product quality model families depending on the context. If a publication appears with a new software product quality model, then it establishes a new model class for which the name of the new software product quality model is introduced. We compute the scores for these software product quality model classes by summing the quality scores of the related individual publications in the cluster. This cumulated score value we call relevance score, which designates how vivid the research interest is in connection with the given software product quality model family. Beside the cumulated score value we also provide the average quality score value and the publication ranges in which the publications appeared. The publication range carries important information showing the period of the publications regarding the given quality model class. It illustrates whether research on the field takes place continually. The relevance score value and the publication range need to be interpreted together.

Moreover, the publications stem not only from the academic research domain but involve companies such as Air France, Siemens, IBM, Samsung, Qualixo and Mitre. Thus, the relevance score value mirrors the industrial interest of the quality model classes to some extent, however, the industrial research interest does not necessarily mean practical everyday use cases. Consequently, we also introduced the average of Google Relative Search Index [47] going back for a 12-month period in the past from the closing date of the manuscript. Google Relative Search Index [47] shows on a continuous scale of [0; 100] how popular the given search strings are in comparison to each other. As search strings we applied the names of the model classes. In some of the cases the quality model class names possess different connotations exceeding the quality domain. Such cases are presented as ”n.a.” values in the ranking table of the identified models.

Kitchenham et al. proposes using quality indicators for each single publication and computing quality scores with regard to several conditions, including also more researchers to compute average scores for increasing reliability of the quality assessment [82], however, they solely utilise these values for the quality assessment of the individual publications and through that for the quality assessment of the review process. Thus, they apply the quality scores in a similar way to our individual quality

scores to assess each publication. However, to the best knowledge of the authors no publication applies the aggregation of these scores according to particular clusters among the research data. This simple technique can be used to express existing differences among the research subjects in a quantitative manner.

By means of this novelty we describe the relevance of the different software product quality model classes from the point of view of the scientific and industrial community as it shows publication intensity and through that research intensity. Some of the quality models were defined in one step and further enhancements were published afterwards. In contrast, some of the quality models were published in smaller increments and not defined fully in one step. This latter approach inherently results in more publications, which we considered while assessing the individual publications as defined in section 2.5.

The background and the problem of the research is illustrated in the section Introduction while the protocol is laid down in the current section.

2.1 Research Questions of the Systematic Literature Review

The research problem can be covered with the single question:

Q1 What software product quality models exist that aim to assess all defined characteristics of software product quality?

2.2 Keyword Development and Search Strategy

In the scope of the keyword development the particularities of the topic area were considered to form search terms that are specific enough to limit the search results on the given field. In the case of Q1, the terminology of the known models were analysed to form the search strings. The ISO/IEC models and their derivatives match with the term ”software product quality model” as this is the terminology of the standard. In contrast, the terminology of the SQALE model is different and less specific as a consequence of which, a broader search string would have been required to identify all the publications automatically. The broader search terms however return such high number of publications that processing each of them would have caused a serious impediment. For this reasons, in accordance with the proposal made by Kitchenham and Brereton [82] automatic and manual searches were integrated. The references of the automatically identified publications pointing at unidentified software product quality models were followed and those publications were also analysed. Depending on the information revealed by the publications gained from the references, new conventional literature review searches were performed, focused on the particular area of interest shown by the referenced publications. The search strings for these manual searches were not recorded and the publications included this way in the scope of the analysis were marked with the origin: ”manual search” in Appendix C. The search strings developed consider the most important synonyms on the field.

2.2.1 Terms for Q1

Search strings developed:

A For ISO/IEC specific models: ”Software Product Quality Model” OR ”Software Product Quality Framework”
B For filtering the topic area in general: (”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement)
C For the field of SQALE: manual search integrated with automatic author search in IEEE

2.2.2 Identifying the Research Databases

Kitchenham and Brereton recommend IEEE and ACM digital libraries on the field of Software Engineering due to their good coverage of important journals and conferences; moreover, at least two general purpose digital libraries from SCOPUS, El Compendix and Web of Science if automated searches are planned with search strings to perform a systematic literature review [82]. The university library of the De Montfort University recommends the following digital libraries for searching Computer Science literature [1]: ACM, IEEE, EBSCO Academic Search Premier, Science Direct, Web of Science. Consolidating the two recommendations, the following digital libraries were selected:

1. ACM Digital Library
2. IEEE
3. EBSCO Academic Search Premier
4. SCOPUS
5. Science Direct
6. Web of Science

In all the databases the search was conducted on the area of Computer Science for the years 2000-2017 on the metadata fields including title, keywords and abstract. In addition, the search was repeated in the ACM and IEEE databases at the end of the manuscript preparation for the period 2018 and 2019 and the outcome is considered in section 3 Results. For the reproducibility of the search all the search strings are documented in the appendix.

2.3 Inclusion and Exclusion Criteria

All documents returned by the automatic search performed with the constructed search strings in the defined databases are included in the review by default unless the exclusion criteria below apply. In addition, quality frameworks with the concepts of existing software product quality models on a field unrelated to computing are introduced in the section 3.1.

[1] http://libguides.library.dmu.ac.uk/c.php?g=51890&p=335386, [accessed: 30.05.2017]

Exclusion Criteria for Documents Identified by the Automatic Search

1. The publication is not a software product quality model (e.g. it is a software process model)
2. The model introduced does not attempt to deal with the whole set of quality properties but focusses on a subset (e.g. models for maintainability or performance)
3. The publication is a comparative study about software product quality models but does not introduce a new software product quality model or the adaptation of an existing model.
4. The publication reports the progress about creating a new software product quality model but the model is not defined at the time of the publication. In contrast, publications defining the enhancements of already available software product quality models are included.
5. The publication highlights only some principles of existing software product quality models without extending or without tailoring a concrete model to a specific application domain.

Exclusion and Inclusion Criteria for the Complementing Manual Searches
The automatic searches were complemented by manual searches as introduced in the previous section 2.2. By following the references cited by the publications identified by the automatic searches resulted also in the identification of software product quality models. In such cases all the exclusion criteria of the automatic search apply. In contrast, if these manually identified publications stem from before the publication year 2000, which was put down in the search strings of the automated searches as a filter criterion, but they introduce software product quality models from which a new software product quality model has been derived since 2000, then they are included in the list of software product quality models identified with a zero relevance score value.

2.4 Extracted Information

The names of the quality models identified, the model descriptions and information to determine whether the quality model is a new model, an adaptation of a known model described by other publications.

It is not always simple to consider the boundaries of already present quality models with regard to published adjustments and tailoring. While doing the classification of the publications, every reasonable effort was made to correctly decide whether the given publication introduces a new quality model or it merely expresses enhancements of an already present model. If the adjustment or tailoring to a specific context of use articulates completely new concepts in comparison to the existing quality model it is built on, then the publication is classified as the introduction of a new software product quality model. In contrast, if an adjustment or tailoring to a specific context of use does not introduce new concepts in the present software product quality model but extends solely its properties, sub-properties or metrics or redefines them in part, then the publication is classified as an adaptation of a present software product quality model not a new model.

2.5 Evaluation Criteria for the Documents

The description of each individual software product quality model differs considerably. Quality models exist with high number of publications, analysis and demonstration of use such as SQALE [99] and the ISO/IEC 9126 [72] standard family. However, other quality models come with concise descriptions and a brief demonstration of use. In addition, a third group of quality model definitions also appear in the identified sources with incomplete descriptions or with the lack of metric definitions. The quality models introduced differ significantly in the amount and the depth of introduction they present.

For assessing the documents returned by the queries, the recommendations in [82, 83] were considered. In this manner for each individual publication we constructed a quality score that shows to which extent the publication is relevant for our purposes.

1. How clear and coherent the publication is?
2. How actual the publication is?

The identified publications were evaluated from two points of views: (1) clarity of presentation and (2) actuality of the publication. Both scoring criteria are presented below in detail. The individual quality score value of each publication is computed by the product of these two factors: presentation clarity and actuality.

2.5.1 Presentation Clarity

Scale: interval
Range: [1;5]
Meaning of the score value: The value indicates how clearly, and completely the publication introduces the software product quality model with regard to concepts and details.

Score value: **5**
- In terms of the software product quality models, the model is presented clearly in appropriate depth including the outline of the concepts of the model, all defined quality properties, sub-properties, possibly metrics, measures and any other defined characteristics necessary to use. If the metrics or measures are not available in the given publication, they must be available elsewhere and the given publication and the documents that publish the measures and metrics must form a consistent unit.

Score value: **0**
- In terms of the software product quality models, (a) the presentation of the model is unclear, (b) the concepts of the model are not outlined, (c) the references to related models are inaccurate, or (d) the model's quality properties, sub-properties metrics, measures or necessary characteristics are not defined nor published elsewhere in an available manner.

If a model is published fully-fledged with its concepts, quality properties, metrics as a consistent unit, with accurate references to other models which influenced its

development, then the maximal value: five is scored. If the model undergoes further adjustments and these developments are published, the adjustments are considered as increments, which do not define a complete model. Consequently, the publications with the increments depending on the depth of information they present usually earn a score less than the maximum.

2.5.2 Actuality of the model

Scale: interval

Range: [1;5]

Meaning of the score value: The value indicates how up-to-date the publication is.

Score value 5: The publication is up-to-date.

Score value 0: The publication is not up-to-date. If the publication has a zero score value, then its quality model is not relevant for the current investigation; however, its concepts can be of importance.

Table 1: Scoring Criteria on Actuality

Publication date	Maximum score value that can be given
2014 or after	5
]2014; 2011]	4
]2011; 2007]	3
]2007; 2004]	2
]2004; 2000]	1
1999 or before	0

2.5.3 Computation of the Quality Score for an Individual Publication

The quality score is computed based on the reviewed materials for each publication. This score can be interpreted as a numeric quality indicator by the product of the two factors (1) publication clarity and (2) actuality of the publication.

The metrics and the guides of the software product quality models if published separately are not counted towards the relevance scores unless they introduce the concepts of the model not published elsewhere. SQUALE publishes two of its model concepts (1) concept of practices [11] and (2) concept of quality defect resolution strategy [68] in detail as part of two separate reports, which mainly define metrics and constitute guides. Consequently, these two publications have been considered for the above reason while computing the scores.

2.6 Recording

The following data about the publications identified were recorded; moreover, the list of publications was processed with regard to the listed criteria:

1. The citation of the publication
2. Content of the publication for analysis
3. Reasons for exclusion if there were any

3 Results

The literature review considers all the software product quality models returned by the documented searches according to the defined inclusion and exclision criteria. The term software product quality framework is used when we emphasize that these models were created with the aim to describe the whole set of software product quality. Software process quality models and cost models are excluded, as well as, product quality models that only aim to handle a certain part of the software product quality but not the whole set. Process quality, product quality and cost models serve different purposes. Kläse et al. published a classification scheme in [86], in which they classify twenty models including process quality, product quality and cost models to provide assistance with quality model selection for a specific purpose, organisation or project.

Each software product quality framework has twofold purpose: (1) to provide means to assesses the quality of a concrete software product and (2) define quality targets for a given software product [67]. The achievement of this purpose depends on the particularities of the software product quality framework.

Table 2 on page 12 and table 3 on page 13 list all the quality model classes identified, with relevance score, average of quality score, publication range, and the 12-month average of Google Relative Search Index [47], sorted by relevance scores to highlight the ones in the focus of vivid research interest in the academic and industrial community. Moreover, all the model classes are listed also separately that were identified by the automatic search. This gives an illustration to which extent the quality models could be identified by the automatic search alone and to which extent the manual search contributed to the final list. Pure automatic search was not efficient enough to identify 14 quality model families. Both tables are also depicted on bar charts with the quality models sorted alphabetically on figure 1 on page 11 and on figure 2 on page 11.

The four indicators: (1) relevance score, (2) quality score average, (3) publication range and (4) the average of the 12-month Google Relative Search Index [47] assist to interpret how far the quality model family is accepted by the scientific and industrial community. The relevance score value encompasses publications and model definitions also from the industrial field such as EMISQ [91], SQUALE [116], FURPS [50, 51], SQAE and ISO9126 combination [25], Quality Model of Kim and Lee [81], which involve research from companies such as Air France, Siemens, Qualixo, IBM, MITRE, and Samsung, but the industrial popularity of the quality model families and their practical, everyday use cases are better approximated with the Google Relative Search Index [47]. The relevance score value is in accordance with the Google Relative Search Index [47] apart from two items in the ranking: FURPS [51] and GQM [134], which means these two quality model families are far more widespread and presumably possess more applications than it could have been assumed based on the publications associated with them. Google Relative Search Index [47] indicates

Table 2: Ranking of the Quality Model Classes by Relevance Scores Including Manual and Automatic Searches

Ranking	Model Class	Relevance Score	Quality Score Average	Publication Range After 2000	Google Relative Search Index, Average for 12 Months
1	ISO25010 [29, 38, 65, 66, 75, 107, 107, 119, 131, 131, 145]	130	16.25	[2000; 2018]	30.02
2	ISO9126 [7, 22, 62, 72, 77, 104, 105, 121, 147]	120	13.33	[2000; 2017]	53.06
3	SQALE [59, 96–102]	107	13.38	[2009; 2016]	18.33
4	Quamoco [44, 149–151]	90	22.5	[2012; 2015]	0
5	EMISQ [91, 124, 125]	38	12.67	[2008; 2011]	0
6	SQUALE [11, 68, 93, 116]	36	9	[2012; 2015]	n.a.
7	ADEQUATE [61, 79]	18	9	[2005; 2009]	n.a.
8	COQUALMO [17, 110]	15	7.5	[2008; 2008]	0.21
=9	FURPS [32, 50, 51]	10	3.33	[2005; 2005]	20.56
=9	SQAE and ISO9126 combination [25]	10	10	[2004; 2004]	0
=9	Ulan et al. [145]	10	10	[2018; 2018]	n.a.
10	Kim and Lee [81]	9	9	[2009; 2009]	n.a.
11	GEQUAMO [43]	5	5	[2003; 2003]	0
12	McCall et al. [14, 114]	1	0.5	[2002; 2002]	n.a.
=13	2D Model [159]	0	0	n.a.	n.a.
=13	Boehm et al. [18]	0	0	n.a.	n.a.
=13	Dromey [30]	0	0	n.a.	n.a.
=13	GQM [134]	0	0	n.a.	40.73
=13	IEEE Metrics Framework Reaffirmed in 2009 [67]	0	0	n.a.	0
=13	Metrics Framework for Mobile Apps [39]	0	0	n.a.	0
=13	SATC [64]	0	0	n.a.	n.a.
=13	SQAE [111]	0	0	n.a.	n.a.
=13	SQUID [84]	0	0	n.a.	n.a.

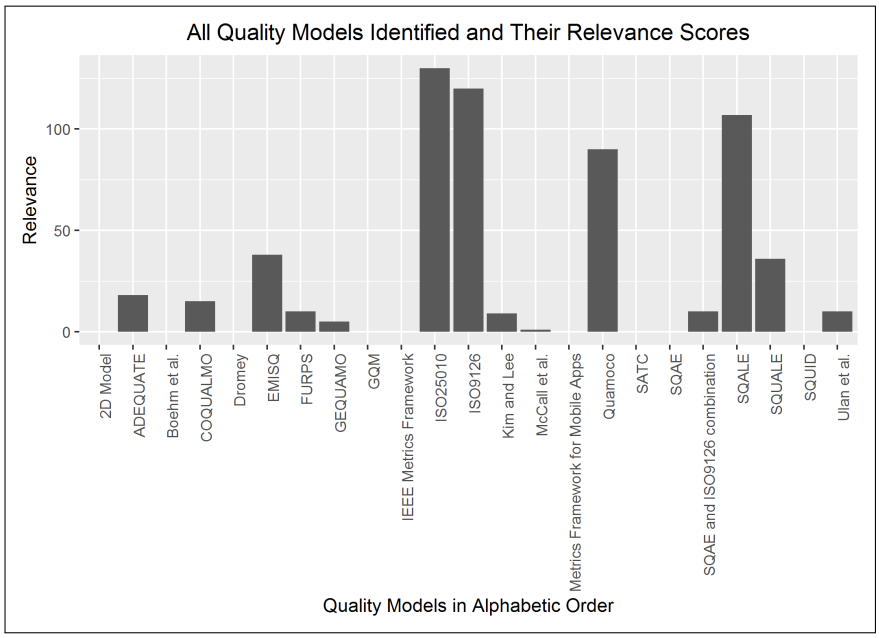


Fig. 1: Relevance Scores of the Quality Model Families Including Manual and Automatic Searches

Table 3: Relevance Scores of the Quality Model Families Excluding Manual Search

No	Ranking	Model Class	Relevance
1	1	ISO25010	95
2	2	ISO9126	75
3	3	Quamoco	45
4	4	EMISQ	20
5	5	Ulan et al.	10
6	6	SQALE	9
7	7	McCall et al.	1
8	=8	Metrics Framework for Mobile Apps	0
9	=8	2D Model	0

more activity for ISO/IEC 9126 [72] than for ISO/IEC 25010 [75], which shows that the practical use cases of ISO/IEC 9126 [72] probably still exceed the applications of its successor standard ISO/IEC 25010 [75] even if the amount of research associated with ISO/IEC 25010 [75] overtook the predecessor standard. The SQALE model [99] achieved the third place in the ranking according to the relevance scores computed on basis of the publications. This result is also in accordance with the Google Relative Search Index [47] if we descope the two outliers: FURPS [51] and GQM [134]. In addition, if SONAR [135], one popular implementation of the SQALE model [99], is involved in the Google Relative Search Index [47], then it suppresses all other search indexes nearly to zero, which means that the most activity in the domain seems to be associated with the widespread implementation of the SQALE model [99]. In addi-

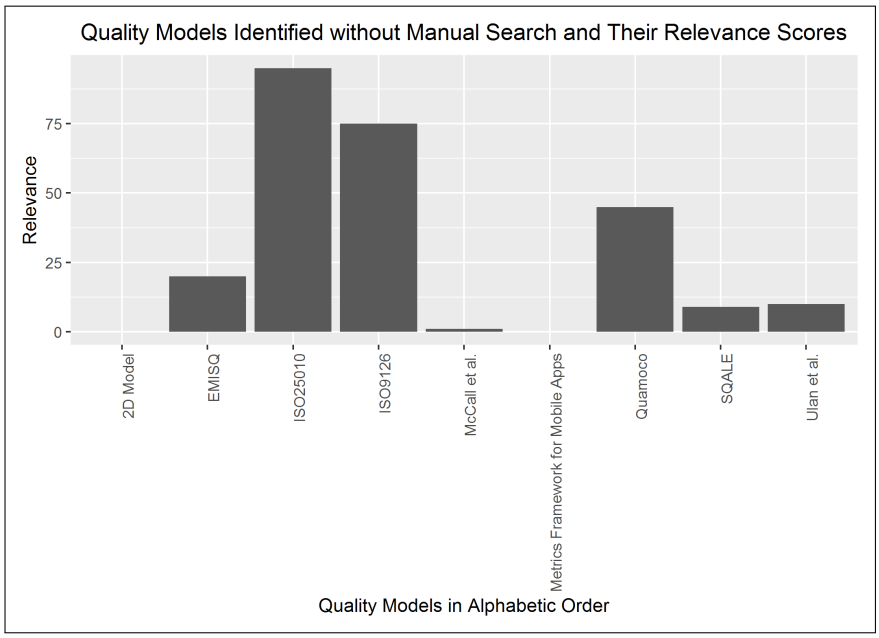


Fig. 2: Relevance Scores of the Quality Model Families Excluding Manual Search

tion, Quamoco [150] and EMISQ [91] quality model families comprise of research but exhibit no search activities, which probably indicates less acceptance in everyday, industrial settings. The quality model families marked with ”n.a.” values in the Google Relative Search Index [47] field indicate that the model name is also associated with other connotations; therefore this indicator cannot be used as measurement for the quality model families. Furthermore, the relevance score of the Quamoco quality model family [44, 149–151] is relatively high but the publication range is narrow 2012-2015, which indicates that the quality model does not show up in the focus of continuous research in a wider community.

As the introduced software product quality models indicate, the endeavour to support the quality assessment by tools and possibly automate it can be observed in the case of all the major quality models [7, 22, 29, 38, 44, 59, 62, 65, 66, 72, 75, 77, 96–102, 104, 105, 119, 121, 147, 149–151]. By the segmentation of quality to internal and external view, only the internal view can be described and assessed by static code analysis. The external view of quality by its definition [72, 75] deals with the software in execution and describes how the product relates to its environment, which cannot fully be automated at present. Thus, it includes dynamic investigations such as testing. Kim and Lee solved to automate the quality assessment based on a model derived from the ISO/IEC 9126 [72] software product quality framework with the internal quality view in 2008 [81]. Furthermore, they integrated the automatic quality assessment into the development lifecycle. Letouzey and Coq created an individual model, SQALE, with the view on the development lifecycle of the software and the integration of quality assessment into this lifecycle [99]. The implementa-

Table 4: Tool Support of the Identified Software Product Quality Model Classes

No.	Software Product Quality Model Classes, Names in Alphabetic Order	Tool Support
1	2D Model [159]	Unknown
2	ADEQUATE [61, 79]	Unknown
3	Boehm et al. [18]	Unknown
4	COQUALMO [17, 110]	Unknown
5	Dromey [30]	Unknown
6	EMISQ [91, 124, 125]	Unknown
7	FURPS [32, 50, 51]	Unknown
8	GEQUAMO [43]	In part
9	GQM [134]	Unknown
10	IEEE Metrics Framework Reaffirmed in 2009 [67]	No
11	ISO25010 [29, 38, 65, 66, 75, 107, 107, 119, 131, 131, 145]	Unknown
12	ISO9126 [7, 22, 62, 72, 77, 104, 105, 121, 147]	Unknown
13	Kim and Lee [81]	In part
14	McCall et al. [14, 114]	Unknown
15	Metrics Framework for Mobile Apps [39]	Unknown
16	Quamoco [44, 149–151]	Yes
17	SATC [64]	Unknown
18	SQAE [111]	Unknown
19	SQAE and ISO9126 combination [25]	Unknown
20	SQALE [59, 96–102]	Yes
21	SQUALE [11, 68, 93, 116]	Yes
22	SQUID [84]	Experimental Toolset
23	Ulan et al. [145]	Unknown

tion of the SQALE model is widespread due to its extensive tool support including Sonar [115, 128, 135, 139, 158]. Nevertheless, SQALE is able to handle the internal quality view only by means of static code analysis. Letouzey and Ilkiewicz anticipates to extend the SQALE model to include architecture analysis in the future [102]. Other quality models such as EMISQ, ADEQUATE require manual intervention to assess quality [61, 79, 91, 124, 125]. In addition, the SQUALE model applies metrics that can be computed automatically and metrics that require manual intervention to measure the external manifestation of quality [93, 116]. If a software product quality model possesses no full automation potential, it can be caused by the broader view of quality it encompasses, i.e. it includes beside the internal view of quality also further manifestations such as the external quality and quality in use. We briefly summarised the tool support for the identified software product quality models in table 4 on page 14.

While conducting the analysis of the identified publication, maintainability and its sub-property analysability was also in focus. It is crucial from the point of view of software quality how quickly an error in a software product can be localised. Execution tracing and logging, used several times as synonyms in the literature, are inevitable mechanisms if errors in distributed, multi-threaded or real-time, embedded applications without user interface have to be located [19, 45, 92, 146]. None of the identified 23 software product quality model classes handle the quality of execution

Table 5: Quality of Execution Tracing or Logging in the Identified Software Product Quality Model Classes

No.	Software Product Quality Model Classes, Names in Alphabetic Order	Execution Tracing or Logging Quality
1	2D Model [159]	No
2	ADEQUATE [61, 79]	No
3	Boehm et al. [18]	No
4	COQUALMO [17, 110]	No
5	Dromey [30]	No
6	EMISQ [91, 124, 125]	No
7	FURPS [32, 50, 51]	No
8	GEQUAMO [43]	No
9	GQM [134]	No
10	IEEE Metrics Framework Reaffirmed in 2009 [67]	No
11	ISO25010 [29, 38, 65, 66, 75, 107, 107, 119, 131, 131, 145]	No but related measures are defined.
12	ISO9126 [7, 22, 62, 72, 77, 104, 105, 121, 147]	No but related metrics are defined.
13	Kim and Lee [81]	No
14	McCall et al. [14, 114]	No
15	Metrics Framework for Mobile Apps [39]	No
16	Quamoco [44, 149–151]	No
17	SATC [64]	No
18	SQAE [111]	No
19	SQAE and ISO9126 combination [25]	No
20	SQALE [59, 96–102]	No
21	SQUALE [11, 68, 93, 116]	Yes, in part.
22	SQUID [84]	No
23	Ulan et al. [145]	No

tracing or logging quality adequately as summarised in table 5 on page 15. SQUALE introduces a quality property to consider, whether trace messages appear on three different severity levels [93, 116], which is only one aspect of execution tracing quality. The ISO/IEC 9126 [72] and ISO/IEC 25010 [75] standards publish analysability metrics and measures which are related to execution tracing quality but they are different [73, 74, 76].

Software quality is an abstract concept. The first software product quality models including the popular ISO/IEC 9126 and ISO/IEC 25010 standards [72, 75] apply hierarchical approach to deal with complexity, i.e. these models decompose abstract entities in a hierarchic manner as far as they become tractable units, then metrics are assigned to those decomposed units to assess each part in a quantitative manner. In addition, the metrics or the scores of decomposed units can be aggregated to express a higher-level score in the hierarchy. Latter frameworks including Quamoco [44, 149–151] create complex meta-models that define the relationship of the internal entities of the quality model. Burgues et al. define a framework to (1) compare and analyse existing quality model approaches, (2) define new models or (3) adapt the existing models to new concepts [20]. They laid down a meta-model for the approach as an extension of the UML meta-model. In Burgues et al.'s framework

definition, distinction is made between two types of entities: (1) generic model to describe the fundamental concepts of quality assessment, and (2) reference model to establish a particular application of the generic model in a specific domain [20]. In addition, they also demonstrate the use of the framework for creating a quality reference model for evaluating external libraries, which can be regarded as a specific case of Commercial-of-the-Shelf (COTS) software selection [20].

Already in the early software product quality models in the 1970's software quality was not described by an aggregated individual indicator but by different high-level quality properties [18, 114]. This trait of the software product quality models remained unchanged during the years of long-term evolution with the exception of Ulan et al.'s model [145]. The reason for developing quality models in this manner is explained by the nature of the different high-level quality properties, which sometimes possess conflicting goals. As a straightforward example the high-level quality properties execution performance and maintainability can be mentioned, which usually emerge in all software product quality models in some form. If the maintainability quality property shows adequate or good values, then the software product possesses a sophisticated execution tracing mechanism, which can log the internal state changes and the routes of the threads of execution within the software. The more accurate information one has in the log files, the easier it is to identify and locate a potential error in the software product. On the other hand, logging all the necessary data about the threads of execution and the internal state changes in the software requires effort during the execution, which deteriorates the performance. For possessing a better execution performance, it would be desirable to deactivate execution tracing. Thus, the quality properties execution performance and maintainability have contradictory quality targets. If a quality model applies hierarchic decomposition to deal with abstraction, then the subordinate quality properties might also have such contradictory quality targets.

3.1 Software Product Quality Models Beyond Software Quality

The hierarchic decomposition process of software product quality applied in the scope of the ISO/IEC 9126 [72] and in its successor ISO/IEC 25010 standards [75] to deal with the abstract notion of quality exceeded software product quality and software technology. The quality frameworks [13, 53, 55, 95, 130, 136, 137] which transferred the principles of the software product quality models to a domain different form software technology are presented in this section.

Bansiya and Davis publish the QMOOD model, which is software technology related even if it is not a software product quality model, in 2002 [13]. The main motivation was to create a quality model for assessing the outcomes of the early phases in the development lifecycle such as analysis and design [13]. The model adapts the principles of the ISO/IEC 9126 standard family [72] and Dromey's model [30]; moreover, it establishes own metrics to measure the object-oriented analysis and design quality [13]. Furthermore, QMOOD offers tool support for automatic evaluation to analyse the source code [13].

Leveraging the guidelines and structures laid down by the ISO/IEC standards, Sproge and Cevere delineate how the quality of study programmes of a Higher Educational Institution can be handled by means of the ISO/IEC 9126 model [136, 137]. Study programmes and software as products share several similar properties from the point of view of quality as both are abstract products [137]. The model they defined possesses five high-level quality characteristics and twenty-one subcharacteristics. The high-level quality characteristics: (1) functionality, (2) usability, (3) efficiency, (4) maintainability, and (5) portability. The Faculty of Information Technology at the Latvia University of Agriculture started to use the model from the academic year 2009/2010. The use of the model was extended to all study programmes at the university from the academic year 2011/2012. The evaluation covered 485 courses of 55 study programmes by 2012.

Gucegliogl u and Demirors implement the quality model of the ISO/IEC 9126 standard [72] for the context of business processes [53]. They found analogy between software products and business processes as both have inputs, outputs, logical structure and operations. Starting form this analogy, they defined four high-level quality characteristics: (1) functionality, (2) reliability, (3) usability, and (4) maintainability. In addition, the model contains a level of subcharacteristics that are linked to the quality metrics of the business processes. By means of this new approach it became feasible to assess business processes of an organisation and to receive early feedback about them.

Gurbuz, Gucegliogl u and Demirors define a quality framework based on the quality model of the ISO/IEC 9126 standard [72] to assess the processes of human resource management. Furthermore, Lepmets, Ras and Renault create a quality model derived from the ISO/IEC 25010 standard [75] to assess service quality [95].

The above quality models stem from the ISO/IEC standard families [72, 75] even if they do not address software product quality at all.

3.2 Evaluation of the Documents Created While Conducting the Current Research

The search process, the evaluation of the documents and preparing the manuscript took more than one year. This period is long enough to miss recent publications with the original end date 2017. Therefore, the automatic searches Q1a and Q1b A were repeated in the two largest computer science archives: (1) IEEE and (2) ACM for the period from January 2018 till the date of finishing the first draft of the manuscript in October 2019. The results of the search were examined and the impact on the trends introduced in the analysis.

Table 6: Repeated Searches and Result Statistics

	IEEE	ACM	Date
Q1aA	0	0	01.01.2018-04.10.2019
Q1bA	9	0	01.01.2018-04.10.2019

Same abbreviations are used as in the appendix D:

Act. Actuality
Clar. Presentation Clarity
ETP Is Execution Tracing Present?
IoT Internet of Things
Ref. Reference
Rel. Relevance
SPQM Software product quality model

Table 7: Evaluation of the Documents Returned by the Q1a and Q1b Searches

Rel.	Act.	Clar.	Model Class	ETP	Model	Ref.
15	5	3	ISO/IEC 25010	no	Assessment of ISO/IEC 25010	[131]
10	5	2	Ulan et al.	no	New quality model identified.	[145]
10	5	2	ISO/IEC 25010	no	ISO/IEC 25010 weighting methods and aggregation of quality properties.	[107]
n.a.	n.a.	n.a.	n.a.	n.a.	Mapping study. Not an SPQM.	[118]
n.a.	n.a.	n.a.	ISO/IEC 25010	n.a.	ISO/IEC 25010 transferred to industrial automation context. Not an SPQM.	[78]
n.a.	n.a.	n.a.	n.a.	n.a.	Mapping study on IoT. Not an SPQM.	[6]
n.a.	n.a.	n.a.	ISO/IEC 9126	n.a.	ISO/IEC 9126 transferred to the software process quality domain. Not an SPQM.	[133]
n.a.	n.a.	n.a.	n.a.	n.a.	Interoperability model for IoT. Not a complete SPQM.	[4]
n.a.	n.a.	n.a.	n.a.	n.a.	Maintainability and reliability metrics. Not an SPQM.	[109]

4 Threats to Validity

The search to identify the sources was performed with two approaches: (1) automatic search in the scientific document databases with defined and recorded search strings, (2) manual search to complement the automatic search as proposed by Kitchenham et al. in [82]. All the relevant documents identified went through a defined scoring process. The documents unrelated to software product quality models were excluded with providing a reason in each case for the exclusion.

Validity can suffer harm if publications with the definition, application, tailoring or research of software product quality models are missed during the search. We endeavoured to minimise this risk by carrying out the automatic searches in six computer science relevant scientific document databases including IEEE and ACM; moreover, the references to software product quality models cited in the identified publications of the automatic search were followed manually and examined.

In addition, the data extraction, the scoring of the publications, and the assignment to the quality model family also include some threat to the validity. This threat we minimised by internal review performed by the authors. Furthermore, we laid down a scoring standard we consistently applied in the course of the research; moreover, we considered individual quality scores, average of the individual quality scores, relevance scores, publication ranges and the 12-month average of the Google Relative Search Index [47]. The individual quality score considers the publication clarity and the actuality of each publication identified; moreover, it also considers that quality models can be published as fully defined models in one step and in smaller increments.

5 Related Works

Hegeman [59] gives a summary on the software product quality models published up to 2011 and investigates also the correlation between SQALE indices [99] and the perceived quality. Galli et al. present quality models which aim to handle all defined characteristics of software product quality and investigate their extension facilities with regard to execution tracing quality in 2013 [40]. On the other hand, Ferenc et al. review the software product quality models up to 2014 [36]. However, (1) the previous research conducted was not implemented as a systematic literature review with defined rigorous rules [83], (2) none of these publications measure the relevance of the presented quality models in the scientific and industrial community (3) the publications do not show the latest developments since 2014. In addition, Hegeman [59] and Ferenc et al. [36] introduce also cost models and partial quality models dealing with a specific part of software product quality only to be used solely for modelling one high-level quality property such as maintainability.

6 Conclusion and Future Work

The main goal of this publication is (1) to collect and classify the quality models that endeavour to handle all manifestations of software product quality and (2) to rank these models from the point of view of the research interest and acceptance in the scientific and industrial community. Large amount of quality models exist but considerably fewer models aim to deal with the whole set of properties of software product quality. On the other hand, the existing models are not of equal relevance however sophisticated the quality model definition is. Both aspects mentioned made necessary to conduct a systematic literature review to discover (1) which quality models exist that aim to deal with the whole set of the quality properties of software products; moreover, (2) a mechanism had to be defined and implemented to measure the relevance of the published software product quality models. We addressed both of these goals in the present research.

We implemented a novel approach that can be applied in systematic literature review processes [83] in general to score each included document in the review and to compute scores for the cluster of related documents to construct an indicator of

relevance for the given research subjects. In our research, each document was scored according to a defined standard and the scores were summed for each software product quality model cluster i.e. software product quality model class. In addition, we considered the average score of the documents in each cluster, the publication range in which the publications appeared, and the average of the 12-month Google Relative Search Index [47]. These indicators assist to determine the relevance in the view of the scientific and industrial community. The publications identified and scored stem not only from academic but also from industrial research involving companies such as Air France, Siemens, IBM, Samsung, Qualixo and Mitre.

The quality models of the ISO/IEC frameworks [72, 75] and the SQALE model [99] stand in the focus of the most vivid research interest. The documents published while this study was being conducted corroborated this trend. The popularity of the ISO/IEC models [72, 75] hide in their simplicity, in the ability of hierarchic decomposition to cope with abstraction and in the potential the models offer for the adaptation to specific project needs. The popularity of the SQALE model [99] is motivated by its quick and simple integration in the software development pipeline and by the automation potential the model provides; however, it solely considers the internal view of quality. SQALE model [99] implementations including Sonar are widespread [115, 128, 135, 139, 158]. Automating the derivatives of the ISO/IEC models [72, 75] is also possible as Kim and Lee introduced [81]. However, only the internal quality view of the framework can be assessed by static code analysis in each case.

SQALE [99] is the only model introduced in our research report which explicitly endeavours to satisfy the representation condition of measurement theory [101]. The representation condition asserts that properties of real world entities measured are mapped in numeric representations in such a way that the numeric representations are equivalent to the reality [99]. Most quality models leverage weighted averages to aggregate quality property values or metrics. Letouzey and Coq point out in [101] that using weighted averages in hierarchies to aggregate the metrics violates the representation condition; moreover, they illustrate the problem by means of use cases with (1) masking effects, (2) compensation effects, and (3) threshold effects for different aggregation operations and show how these effects hide the underlying values in the hierarchy [101]. Consequently, SQALE [99] applies a common scale for measuring each quality property and only the sum operation is used to avoid the mentioned anomalies.

The simplicity of the ISO/IEC quality models [72, 75] and its ability to tackle abstraction provided a starting point for researchers who transferred the same concepts to completely different application domains including teaching quality in Higher Education and service quality [53, 55, 95, 136, 137].

Wagner et al. expresses the following requirements towards any quality model to be able to assess quality in an appropriate manner in 2012 [150]: (1) it must be interpretable for decision makers, (2) it must be able to handle incomplete information, (3) it must allow for contradictory quality aspects. These requirements set out the way for the application of fuzzy logic in the domain. Fuzzy logic makes possible to consider contradictory quality targets, incomplete information; moreover, it

makes possible to describe the quality model with linguistic rules, which eases the interpretation [154–157].

Furthermore, the present study verified that existing software product quality models do not adequately assess the quality of logging and execution tracing, however, it would be required for constructing a maintainability-analysability quality property [41, 42]. The authors will endeavour to satisfy this demand in their future works by means of applying computational intelligence including fuzzy logic to represent abstraction, incomplete information, and conflicting quality properties.

7 Conflict of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. 2014 world congress on computer applications and information systems, WCCAIS 2014. In: 2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014. Institute of Electrical and Electronics Engineers Inc. (2014)
2. Proceedings - 2016 10th international conference on the quality of information and communications technology, quatic 2016. In: Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016. Institute of Electrical and Electronics Engineers Inc. (2017)
3. Proceedings - 26th international workshop on software measurement, iwsm 2016 and the 11th international conference on software process and product measurement, mensura 2016. In: Proceedings - 26th International Workshop on Software Measurement, IWSM 2016 and the 11th International Conference on Software Process and Product Measurement, Mensura 2016. Institute of Electrical and Electronics Engineers Inc. (2017)
4. Abdelouahid, R.A., Marzak, A.: Towards a new interoperability quality model for iots. In: 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), pp. 1–6 (2018). DOI 10.1109/ISIICT.2018.8613289
5. Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R.: Investigating effect of design metrics on fault proneness in object-oriented systems. Journal of Object Technology **6**(10), 127–141 (2007)
6. Ahmed, B.S., Bures, M., Frajtak, K., Cerny, T.: Aspects of quality in internet of things (iot) solutions: A systematic mapping study. IEEE Access **7**, 13758–13780 (2019). DOI 10.1109/ACCESS.2019.2893493
7. Andreou, A.S., Tziakouris, M.: A quality framework for developing and evaluating original software components. Information and Software Technology **49**(2), 122–141 (2007). DOI 10.1016/j.infsof.2006.03.007
8. Atterzadeh, I., Ow, S.H.: Ca novel soft computing model to increase the accuracy of software development cost estimation. In: Proceedings of the 2nd International Conference on Computer and Automation Engineering, pp. 603–607 (2010)
9. Bakota, T., Hegedus, P., Kortvelyesi, P., Ferenc, R., Gyimothy, T.: A probabilistic software quality model. In: 27th IEEE International Conference on Software Maintenance (ICSM) (2011). DOI 10.1109/ICSM.2011.6080791
10. Bakota, T., Hegedus, P., Ladanyi, G.: A cost model based on software maintainability. In: 28th IEEE International Conference on Software Maintenance (ICSM) (2012). DOI 10.1109/ICSM.2012.6405288
11. Balmas, F., Bellingard, F., Denier, S., Ducasse, S., Franchet, B., Laval, J., Mordal-Manet, K., Vaillergues, P.: Practices in the Squale quality model (squale deliverable 1.3). [Online], October, 2010, [Accessed: 16.11.2017] (2010). URL `http://www.squale.org/quality-models-site/research-deliverables/WP1.3Practices-in-the-Squale-Quality-Modelv2.pdf`

12. Balmas, F., Bergel, A., Bellingard, F., Denier, S., Ducasse, S., Laval, J., Mordal-Manet, K., Abdeen, H., Bellingard, F.: Software metric for java and c++ workpackage: 1.1 version: 2. [Online], March, 2010, [Accessed: 16.11.2017] (2010). URL http://www.squale.org/quality-models-site/research-deliverables/WP1.1Software-metrics-for-Java-and-Cpp-practicesv2.pdf
13. Bansiya, J., Davis, C.G.: A hierarchical model for object-oriented design quality assessment. IEEE Transactions on Software Engineering **28**(1), 4–17 (2002). DOI 10.1109/32.979986
14. Benedicenti, L., Wang, V.W., Paranjape, R.: A quality assessment model for java code. In: Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 687–690 (2002)
15. Berry, M., Johnson, C.S.: Improving the quality of information for software project management, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4895 LNCS (2008)
16. Berry, M., Johnson, C.S.: Software Process and Product Measurement, chap. Improving the Quality of Information for Software Project Management, pp. 1–20. Springer-Verlag, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-85553-81. URL http://dx.doi.org/10.1007/978-3-540-85553-81
17. Boehm, B., Chulani, S.: Modeling software defect introduction and removal – COQUALMO (constructive quality model). Tech. rep., USC-CSE Technical Report (1999)
18. Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative evaluation of software quality. In: Proceedings of the 2nd International Conference on Software Engineering (1976)
19. Buch, I., Park, R.: Improve debugging and performance tuning with ETW. MSDN Magazine, [Online], [Accessed: 01.01.2012], http://msdn.microsoft.com/en-us/magazine/cc163437.aspx (2007)
20. Burgués, X., Franch, X., Ribó, J.M.: A MOF-compliant approach to software quality modeling, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3716 LNCS (2005)
21. Coq, T.: Verification and validation in the recommended practice for integrated software-dependent systems. In: First International Conference on Advances in System Testing and Validation Lifecycle, pp. 57–61 (2009). DOI 10.1109/VALID.2009.36
22. Correia, J., Visser, J.: Certification of technical quality of software products. In: International Workshop on Foundations and Techniques for Open Source Software Certification, pp. 35–51 (2008)
23. Correia, J.P., Kanellopoulos, Y., Visser, J.: A survey-based study of the mapping of system properties to iso/iec 9126 maintainability characteristics. In: 2009 IEEE International Conference on Software Maintenance, pp. 61–70 (2009). DOI 10.1109/ICSM.2009.5306346
24. Côté, M.A., Suryn, W., Georgiadou, E.: In search for a widely applicable and accepted software quality model for software quality engineering. Software Quality Journal **15**(4), 401–416 (2007). DOI 10.1007/s11219-007-9029-0
25. Côté, M.A., Suryn, W., Martin, R.A., Laporte, C.Y.: Evolving a corporate software quality assessment exercise: A migration path to ISO/IEC 9126. Software Quality Professional **6**(3), 4–17 (2004)
26. Davila, A., Melendez, K., Flores, L.: Establishing software product quality requirements according to international standards. IEEE Latin America Transactions **4**(2), 100–106 (2006). DOI 10.1109/TLA.2006.1642457
27. Deissenboeck, F., Heinemann, L., Herrmannsdoerfer, M., Lochmann, K., Wagner, S.: The quamoco tool chain for quality modeling and assessment. In: Proceedings of the 33rd International Conference on Software Engineering, ICSE '11, pp. 1007–1009. ACM, New York, NY, USA (2011). DOI 10.1145/1985793.1985977
28. Deissenboeck, F., Juergens, E., Lochmann, K., Wagner, S.: Software quality models: Purposes, usage scenarios and requirements. In: 2009 ICSE Workshop on Software Quality, pp. 9–14 (2009)
29. Domínguez-Mayo, F.J., Escalona, M.J., Mejías, M., Ross, M., Staples, G.: Quality evaluation for model-driven web engineering methodologies. Information and Software Technology **54**(11), 1265–1282 (2012). DOI //doi.org/10.1016/j.infsof.2012.06.007
30. Dromey, R.: A model for software product quality. In: IEEE Transactions on Software Engineering, vol. 21, pp. 146–162 (1995)
31. Ducasse, S., Denier, S., Balmas, F., Bergel, A., Laval, J., Mordal-Manet, K., Bellingard, F.: Visualization of practices and metrics, workpackage: 1.2 version: 1.1. [Online], March, 2010, [Accessed: 16.11.2017] (2010). URL http://www.squale.org/quality-models-site/research-deliverables/WP1.2Visualization-of-Practices-and-Metricsv1.1.pdf
32. Eeles, P.: Capturing architectural requirements. Online, [Accessed: 19.04.2018] (2005). URL https://www.ibm.com/developerworks/rational/library/4706-pdf.pdf
33. Eeles, P.: Capturing architectural requirements. IBM [Online], [Accessed: 16.11.2018] (2005). URL https://www.ibm.com/developerworks/rational/library/4706-pdf.pdf

34. Ericsson, M., Löwe, W., Olsson, T., Toll, D., Wingkvist, A.: A study of the effect of data normalization on software and information quality assessment. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), vol. 2, pp. 55–60 (2013). DOI 10.1109/APSEC.2013.112
35. Falessi, D., Sabetzadeh, M., Briand, L., Turella, E., Coq, T., Panesar-Walawege, R.K.: Planning for safety standards compliance: A model-based tool-supported approach. IEEE Software **29**(3), 64–70 (2012). DOI 10.1109/MS.2011.116
36. Ferenc, R., Hegedüs, P., Gyimóthy, T.: Software Product Quality Models, Chapter. In book: Evolving Software Systems. Springer, Berlin, Heidelberg (2014). DOI 10.1007/978-3-642-45398-43
37. Foley, O., Helfert, M.: Information quality and accessibility. In: Innovations and Advances in Computer Sciences and Engineering, pp. 477–481 (2010). DOI 10.1007/978-90-481-3658-284
38. Forouzani, S., Chiam, Y.K., Forouzani, S.: Method for assessing software quality using source code analysis. In: ACM International Conference Proceeding Series, pp. 166–170. Association for Computing Machinery (2016). DOI 10.1145/3033288.3033316
39. Franke, D., Weise, C.: Providing a software quality framework for testing of mobile applications. In: Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation, ICST 2011, pp. 431–434 (2011). DOI 10.1109/ICST.2011.18
40. Galli, T.: Fuzzy logic based software product quality model for execution tracing. MPhil Thesis, Centre for Computational Intelligence, De Montfort University, Leicester, UK, [Online], 2013, [Accessed: 05.02.2018]. URL https://www.dora.dmu.ac.uk/bitstream/handle/2086/9736/MPhilThesisTamasGalli2013ExaminedFinal.pdf
41. Galli, T., Chiclana, F., Carter, J., Janicke, H.: Modelling execution tracing quality by type-1 fuzzy logic. Acta Polytechnica Hungarica **8**(10), 49–67 (2013). DOI 10.12700/APH.10.08.2013.8.3
42. Galli, T., Chiclana, F., Carter, J., Janicke, H.: Towards introducing execution tracing to software product quality frameworks. Acta Polytechnica Hungarica **11**(3), 5–24 (2014). DOI 10.12700/APH.11.03.2014.03.1
43. Georgiadou, E.: GEQUAMO — a generic, multilayered, customisable, software quality model. Software Quality Journal **11**(4), 313–323 (2003). DOI 1025817312035
44. Gleirscher, M., Golubitskiy, D., Irlbeck, M., Wagner, S.: Introduction of static quality analysis in small- and medium-sized software enterprises: experiences from technology transfer. Software Quality Journal **22**(3), 499–542 (2014). DOI 10.1007/s11219-013-9217-z
45. Godefroid, P., Nagappan, N.: Concurrency at Microsoft – an exploratory survey. [Online], [Accessed: 12.03.2019] (2007). URL https://patricegodefroid.github.io/publicpsfiles/ec2.pdf
46. Gong, J., Lu, J., Cai, L.: An induction to the development of software quality model standards. In: 2016 Third International Conference on Trustworthy Systems and their Applications (TSA), pp. 117–122 (2016). DOI 10.1109/TSA.2016.28
47. Google: Google search trends for the past 12 months, worldwide. [Online], [Accessed: 17.02.2020] (2020). URL https://trends.google.com/trends/explore
48. Grady, R.: Practical Software Metrics for Project Management and Process Improvement. Prentice Hall (1992)
49. Grady, R., Caswell, D.: Software Metrics: Establishing a Company-Wide Program. Prentice Hall (1987)
50. Grady, R.B.: Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, USA, NJ (1992)
51. Grady, R.B., Caswell, D.L.: Software Metrics: Establishing a Company-wide Program. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1987)
52. Gronewold, A.D., Borsuk, M.E.: A software tool for translating deterministic model results into probabilistic assessments of water quality standard compliance. Environmental Modelling and Software **24**(10), 1257–1262 (2009). DOI //doi.org/10.1016/j.envsoft.2009.04.004
53. Gucegliooglu, A.S., Demirors, O.: A process based model for measuring process quality attributes, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3792 LNCS (2005). DOI 10.1007/1158601212
54. Gulezian, R.: Software quality measurement and modeling, maturity, control and improvement. In: Proceedings of Software Engineering Standards Symposium, pp. 52–59 (1995). DOI 10.1109/SESS.1995.525951
55. Gurbuz, O., Gucegliooglu, A.S., Demirors, O.: Application of process quality measurement frameworks for human resource management processes. In: 2011 IEEE International Conference on Quality and Reliability, pp. 426–430 (2011). DOI 10.1109/ICQR.2011.6031754
56. Hamdan, S., Alramouni, S.: A quality framework for software continuous integration. Procedia Manufacturing **3**, 2019–2025 (2015). DOI 10.1016/j.promfg.2015.07.249

57. Hegedus, P.: A probabilistic quality model for c# — an industrial case study. Acta Cybernetica **21**(1), 135–147 (2013)
58. Hegedus, P.: Revealing the effect of coding practices on software maintainability. In: 29th IEEE International Conference on Software Maintenance (ICSM) (2013). DOI 10.1109/ICSM.2013.99
59. Hegeman, J.H.: On the quality of quality models. MSc Thesis, University Twente, [Online], [Accessed: 16.11.2018] (2011). URL https://essay.utwente.nl/61040/1/MScJHegeman.pdf
60. Heitlager, I., Kuipers, T., , Visser, J.: A practical model for measuring maintainability. In: Proceedings of the Quality of Information and Communications Technology QUATIC 2007, p. 30–39 (2007)
61. Horgan, G., Khaddaj, S.: Use of an adaptable quality model approach in a production support environment. The Journal of Systems and Software **82**(4), 730–738 (2009). DOI 10.1016/j.jss.2008.10.009
62. Hu, W., Loeffler, T., Wegener, J.: Quality model based on ISO/IEC 9126 for internal quality of matlab/simulink/stateflow models. In: 2012 IEEE International Conference on Industrial Technology, pp. 325–330 (2012). DOI 10.1109/ICIT.2012.6209958
63. Hughes, T.M.: SAS® Data Analytic Development: Dimensions of Software Quality, pp. 1–606. SAS® Data Analytic Development: Dimensions of Software Quality. wiley (2016). DOI 10.1002/9781119255680
64. Hyatt, L.E., Rosenberg, L.H.: A software quality model and metrics for identifying project risks and assessing software quality. In: Proceedings of Product Assurance Symposium and Software Product Assurance Workshop, EAS SP-377, European Space Agency (1996)
65. Idri, A., Bachiri, M., Fernandez-Aleman, J.L., Toval, A.: Experiment design of free pregnancy monitoring mobile personal health records quality evaluation. pp. 1–6. IEEE (2016). DOI 10.1109/HealthCom.2016.7749501
66. Idri, A., Bachiri, M., Fernández-Alemán, J.L.: A framework for evaluating the software product quality of pregnancy monitoring mobile personal health records. Journal of medical systems **40**(3), 1–17 (2016). DOI 10.1007/s10916-015-0415-z
67. IEEE Computer Society: IEEE Stdandard 1061-1998: IEEE Standard for a Software Quality Metrics Methodology. IEEE (1998)
68. INRIA RMoD, Paris 8, Qualixo: Technical model for remediation (workpackage 2.2), [online], [accessed: 16.11.2017] (2010). URL http://www.squale.org/quality-models-site/research-deliverables/WP2.2Technical-Model-for-Remediationv1.pdf
69. International Organization for Sandardization: ISO/IEC 25020:2007, software engineering - software product quality requirements and evaluation (SQauRE) - measurement reference model and guide (2007)
70. International Organization for Sandardization: ISO/IEC 25021:2012, software engineering - software product quality requirements and evaluation (SQauRE) - quality measure elements (2012)
71. International Organization for Sandardization: ISO/IEC 25022:2016, software engineering - software product quality requirements and evaluation (SQauRE) - measurement of quality in use (2016)
72. International Organization for Standardization: ISO/IEC 9126-1:2001, software engineering – product quality – part 1: Quality model (2001)
73. International Organization for Standardization: ISO/IEC TR 9126-2:2003, software engineering – product quality – part 2: External metrics (2003)
74. International Organization for Standardization: ISO/IEC TR 9126-3:2003, software engineering – product quality – part 3: Internal metrics (2003)
75. International Organization for Standardization: ISO/IEC 25010:2011, systems and software engineering – systems and software quality requirements and evaluation (SQuaRE) – system and software quality models (2011)
76. International Organization for Standardization: ISO/IEC 25023:2016, systems and software engineering – systems and software quality requirements and evaluation (SQuaRE) – measurement of system and software product quality (2016)
77. Kanellopoulos, Y., Tjortjis, C., Heitlager, I., Visser, J.: Interpretation of source code clusters in terms of the ISO/IEC-9126 maintainability characteristics. In: Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR, pp. 63–72 (2008). DOI 10.1109/CSMR.2008.4493301
78. Karnouskos, S., Sinha, R., Leitão, P., Ribeiro, L., Strasser, T.I.: Assessing the integration of software agents and industrial automation systems with iso/iec 25010. In: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 61–66 (2018). DOI 10.1109/INDIN.2018.8471951

79. Khaddaj, S., Horgan, G.: A proposed adaptable quality model for software quality assurance. Journal of Computer Science **1**(4), 482–487 (2005). DOI 10.3844/jcssp.2005.482.487
80. Khoshgoftaar, T.M., Allen, E.B.: Multivariate assessment of complex software systems: a comparative study. In: Engineering of Complex Computer Systems, 1995. Held jointly with 5th CSESAW, 3rd IEEE RTAW and 20th IFAC/IFIP WRTP, Proceedings., First IEEE International Conference on, pp. 389–396 (1995). DOI 10.1109/ICECCS.1995.479364
81. Kim, C., Lee, K.: Software quality model for consumer electronics product. In: Proceedings of the 9th International Conference on Quality Software, pp. 390–395 (2008)
82. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. Information and Software Technology **55**, 2049–2075 (2013)
83. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical Report, EBSE-2007-01 (2007)
84. Kitchenham, B., Linkman, S., Pasquini, A., Nanni, V.: The SQUID approach to defining a quality model. Software Quality Journal **6**(3), 211–233 (1997). DOI 1018516103435
85. Kitchenham, B., Pfleeger, S.: Software quality: the elusive target. IEEE Software **13**(1), 12–21 (1996)
86. Kläs, M., Heidrich, J., Münch, J., Trendowicz, A.: CQML Scheme: A classification scheme for comprehensive quality model landscapes. In: 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, pp. 243–250 (2009). DOI 10.1109/SEAA.2009.88
87. Kläs, M., Lampasona, C., Münch, J.: Adapting software quality models: Practical challenges, approach, and first empirical results. In: 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 341–348 (2011). DOI 10.1109/SEAA.2011.62
88. Kläs, M., Lampasona, C., Nunnenmacher, S., Wagner, S., Herrmannsdörfer, M., Lochmann, K.: How to evaluate meta-models for software quality? In: Proceedings of the Joined International Conferences IWSM/ MetriKon/ Mensura, pp. 443–462 (2010)
89. Knauss, E., Boustani, C.E.: Assessing the quality of software requirements specifications. In: 2008 16th IEEE International Requirements Engineering Conference, pp. 341–342 (2008). DOI 10.1109/RE.2008.29
90. Konopka, B.M., Nebel, J.C., Kotulska, M.: Quality assessment of protein model-structures based on structural and functional similarities. BMC Bioinformatics **13**(1) (2012). DOI 10.1186/1471-2105-13-242
91. Kothapalli, C., Ganesh, S.G., Singh, H.K., Radhika, D.V., Rajaram, T., Ravikanth, K., Gupta, S., Rao, K.: Continual monitoring of code quality. In: Proceedings of the 4th India Software Engineering Conference 2011, ISEC'11, pp. 175–184 (2011). DOI 10.1145/1953355.1953379
92. Laddad, R.: AspectJ in Action. Manning, Second Edition (2009)
93. Laval, J., Bergel, A., Ducasse, S.: Assessing the quality of your software with MoQam. Online, [Accessed: 06.03.2018] (2008). URL https://hal.inria.fr/inria-00498482
94. Lehman, M., Ramil, J.: Rules and tools for software evolution planning and management. Annals of Software Engineering, Special Issue on Software Management **11**(1), 15–44 (2001)
95. Lepmets, M., Ras, E., Renault, A.: A quality measurement framework for IT services. In: Proceedings - 2011 Annual SRII Global Conference, SRII 2011, pp. 767–774 (2011). DOI 10.1109/SRII.2011.84
96. Letouzey, J., Coq, T.: The SQALE models for assessing the quality of real time source code. Online, [Accessed: 17.07.2017] (2010). URL https://pdfs.semanticscholar.org/4dd3/a72d79eb2f62fe04410106dc9fcc27835ce5.pdf?ga=2.24224186.1861301954.1500303973-1157276278.1497961025
97. Letouzey, J.L.: The SQALE method for evaluating technical debt. In: Third International Workshop on Managing Technical Debt (MTD), pp. 31–36 (2012). DOI 10.1109/MTD.2012.6225997
98. Letouzey, J.L.: Managing large application portfolio with technical debt related measures. In: Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), p. 181 (2016). DOI 10.1109/IWSM-Mensura.2016.035
99. Letouzey, J.L.: The SQALE method for managing technical debt, definition document v1.1. [Online], [Accessed: 02.08.2017] (2016). URL http://www.sqale.org/wp-content/uploads//08/SQALE-Method-EN-V1-1.pdf
100. Letouzey, J.L., Coq, T.: The SQALE quality and analysis models for assessing the quality of ada source code. Online, [Accessed: 17.07.2017] (2009). URL http://www.adalog.fr/publicat/sqale.pdf

101. Letouzey, J.L., Coq, T.: The SQALE analysis model: An analysis model compliant with the representation condition for assessing the quality of software source code. In: 2010 Second International Conference on Advances in System Testing and Validation Lifecycle, pp. 43–48 (2010)
102. Letouzey, J.L., Ilkiewicz, M.: Managing technical debt with the SQALE method. IEEE Software **29**(6), 44–51 (2012). DOI 10.1109/MS.2012.129
103. Li, J., Skramstad, T., Coq, T.: Interface information management tools for the maritime and oil and gas industry. In: 2015 IEEE 39th Annual Computer Software and Applications Conference,, pp. 164–169 (2015). DOI 10.1109/COMPSAC.2015.227
104. Li, Y., Man, Z.: A fuzzy comprehensive quality evaluation for the digitizing software of ethnic antiquarian resources. In: 2008 International Conference on Computer Science and Software Engineering, vol. 5, pp. 1271–1274 (2008). DOI 10.1109/CSSE.2008.304
105. Liang, S.K., Lien, C.T.: Selecting the optimal ERP software by combining the ISO 9126 standard and fuzzy AHP approach. Contemporary Management Research **3**(1), 23 (2006). DOI 10.7903/cmr.10
106. Lincke, R., Lundberg, J., Löwe, W.: Comparing software metrics tools. In: ISSTA'08: Proceedings of the 2008 International Symposium on Software Testing and Analysis 2008, pp. 131–141 (2008). DOI 10.1145/1390630.1390648
107. Liu, X., Zhang, Y., Yu, X., Liu, Z.: A software quality quantifying method based on preference and benchmark data. In: 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 375–379 (2018). DOI 10.1109/SNPD.2018.8441145
108. Liu, Z., Liu, T., Lu, T., Cai, L., Yang, G.: Agent-based online quality measurement approach in cloud computing environment. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 1, pp. 686–690 (2010). DOI 10.1109/WI-IAT.2010.213
109. Ludwig, J., Xu, S., Webber, F.: Static software metrics for reliability and maintainability. In: 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), pp. 53–54 (2018)
110. Madachy, R., Boehm, B.: Assessing Quality Processes with ODC COQUALMO, *Making Globally Distributed Software Development a Success Story*, vol. 5007, pp. 198–209. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-79588-918
111. Martin, R.A., Shafer, L.H.: Providing a framework for effective software quality assessment—a first step in automating assessments. In: Proceedings of the first annual software engineering and economics conference (1996)
112. Mayr, A., osch, R.P., Saft, M.: Objective measurement of safety in the context of iec 61508-3. In: Proceedings of the 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, SEAA '13, pp. 45–52. IEEE Computer Society, Washington, DC, USA (2013). DOI 10.1109/SEAA.2013.32. URL http://dx.doi.org/10.1109/SEAA.2013.32
113. Mayr, A., Plösch, R., Saft, M.: Towards an operational safety standard for software: Modelling iec 61508 part 3. In: Proceedings - 18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, ECBS 2011, pp. 97–104 (2011). DOI 10.1109/ECBS.2011.8
114. McCall, J.A., Richards, P.K., Walters, G.F.: Factors in software quality, concept and definitions of software quality. [Online], [Accessed: 06.03.2018] (1977). URL http://www.dtic.mil/dtic/tr/fulltext/u2/a049014.pdf
115. Mia Software: Mia Quality. [Online], [Accessed: 16.02.2018] (2017). URL http://www.mia-software.com/produits/mia-quality/
116. 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. [Online], [Accessed: 06.03.2018] (2009). URL https://hal.inria.fr/inria-00637364
117. Narman, P., Johnson, P., Nordstrom, L.: Enterprise architecture: A framework supporting system quality analysis. In: Proceedings of the IEEE International Annual Enterprise Distributed Object Computing Conference EDOC, p. 63–72 (2007)
118. Nistala, P., Nori, K.V., Reddy, R.: Software quality models: A systematic mapping study. In: 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), pp. 125–134 (2019). DOI 10.1109/ICSSP.2019.00025
119. Ouhbi, S., Idri, A., Fernández-Alemán, J.L., Toval, A., Benjelloun, H.: Applying ISO/IEC 25010 on mobile personal health records. In: HEALTHINF 2015 - 8th International Conference on Health Informatics, Proceedings; Part of 8th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2015, pp. 405–412. SciTePress (2015)
120. Panesar-Walawege, R.K., Sabetzadeh, M., Briand, L., Coq, T.T.: Characterizing the chain of evidence for software safety cases: A conceptual model based on the iec 61508 standard. In: 2010 Third

International Conference on Software Testing, Verification and Validation, pp. 335–344 (2010). DOI 10.1109/ICST.2010.12
121. Parthasarathy, S., Sharma, S.: Impact of customization over software quality in ERP projects: an empirical study. Software Quality Journal **25**(2), 581–598 (2017). DOI 10.1007/s11219-016-9314-x
122. Perry, W.E.: Quality Assurance for Information Systems: Method, Tools and Techniques. John Wiley and Sons (1991)
123. Pitula, K.: On requirements elicitation for software projects in ict for development. Ph.D. thesis (2010). AAINR71106
124. Plösch, R., Gruber, H., Hentschel, A., Körner, C., Pomberger, G., Schiffer, S., Saft, M., Storck, S.: The EMISQ method and its tool support-expert-based evaluation of internal software quality. Innovations in Systems and Software Engineering **4**(1), 3–15 (2008). DOI 10.1007/s11334-007-0039-7
125. Plösch, R., Gruber, H., Körner, C., Saft, M.: A method for continuous code quality management using static analysis. In: 2010 Seventh International Conference on the Quality of Information and Communications Technology, pp. 370–375 (2010). DOI 10.1109/QUATIC.2010.68
126. Rahman, A.A., Sahibuddin, S., Ibrahim, S.: A unified framework for software engineering process improvement - a taxonomy comparative analysis. In: 2011 5th Malaysian Conference in Software Engineering, MySEC 2011, pp. 153–158 (2011). DOI 10.1109/MySEC.2011.6140661
127. Rahman, A.A., Sahibuddin, S., Ibrahim, S.: A taxonomy analysis for multi-model process improvement from the context of software engineering processes and services. International Journal of Digital Content Technology and its Applications **6**(22), 56–65 (2012). DOI 10.4156/jdcta.vol6.issue22.6
128. Security Reviewer Srl: Security Reviewer. [Online], [Accessed: 16.02.2018] (2017). URL `http://www.securityreviewer.net/`
129. Seffah, A., Kececi, N., Donyaee, M.: QUIM: a framework for quantifying usability metrics in software quality models. In: Proceedings Second Asia-Pacific Conference on Quality Software, pp. 311–318 (2001). DOI 10.1109/APAQS.2001.990036
130. Shatnawi, R., Li, W.: An empirical assessment of refactoring impact on software quality using a hierarchical quality model. International Journal of Software Engineering and its Applications **5**(4), 127–150 (2011)
131. Shen, P., Ding, X., Ren, W., Yang, C.: Research on software quality assurance based on software quality standards and technology management. In: 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 385–390 (2018). DOI 10.1109/SNPD.2018.8441142
132. Shubhamangala, B.R., Suma, V., Reddy, P.A., Singh, C.G.: Quality attribute focused multilayer requirement elicitation: Judicious approach to drive business value. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2069–2075 (2013). DOI 10.1109/ICACCI.2013.6637500. ID: 10
133. Simão Monteiro, S.B., Fernandes Lima, A.C., Cristina Venturini, F., de Oliveira, W.S.: Continuous improvement of systems in maintenance using a proactive quality management. In: 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), pp. 47–55 (2018). DOI 10.1109/QUATIC.2018.00017
134. van Solingen, R., Berghout, E.: The Goal/Question/Metric Method a practical guide for quality improvement of software development. McGraw Hill Publishing, England (1999)
135. SonarSource: SonarQube. [Online], [Accessed: 16.02.2018] (2017). URL `https://www.sonarqube.org`
136. Sproge, S.: Evaluation of study programme external quality. Research for Rural Development 2011, Vol 1 pp. 179–185 (2011)
137. Sproge, S., Cevere, R.: Assessment of study programme quality at higher education institution. Engineering for Rural Development - International Scientific Conference **11**, 663 (2012)
138. SQUALE: Final version of the spreadsheet tool used to estimate the return of investment of a global quality process [in french], workpackage: 2.3. [Online], September 2010, [Accessed: 16.11.2017] (2010). URL `http://www.squale.org/quality-models-site/research-deliverables/WP2.3ROI-estimationv1.xls`
139. Squoring Technologies SAS: Squoring. [Online], [Accessed: 16.02.2018] (2017). URL `https://www.squoring.com/`
140. Staron, M., Meding, W., Niesel, K., Abran, A.: A key performance indicator quality model and its industrial evaluation. In: 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 170–179 (2016). DOI 10.1109/IWSM-Mensura.2016.033

141. of Stuttgart, U.: Quamoco open quality model. Online, [Accessed: 15.12.2018] (2012). URL http://www.quamoco.de/
142. of Stuttgart, U.: Quamoco tool support for quality modelling and evaluation. Online, [Accessed: 15.12.2018] (2012). URL http://www.quamoco.de/webmodel/home.html
143. Szabo, R.M., Khoshgoftaar, T.M.: An assessment of software quality in a c++ environment. In: Software Reliability Engineering, 1995. Proceedings., Sixth International Symposium on, pp. 240–249 (1995). DOI 10.1109/ISSRE.1995.497663
144. Tekinerdogan, B., Ali, N., Grundy, J., Mistrik, I., Soley, R.: Quality concerns in large-scale and complex software-intensive systems, pp. 1–17. Software Quality Assurance: In Large Scale and Complex Software-intensive Systems. Elsevier Inc. (2015). DOI 10.1016/B978-0-12-802301-3.00001-6
145. Ulan, M., Hönel, S., Martins, R.M., Ericsson, M., Löwe, W., Wingkvist, A., Kerren, A.: Quality models inside out: Interactive visualization of software metrics by means of joint probabilities. In: 2018 IEEE Working Conference on Software Visualization (VISSOFT), pp. 65–75 (2018). DOI 10.1109/VISSOFT.2018.00015
146. Uzelac, V., Milenkovic, A., Burtscher, M., Milenkovic, M.: Real-time unobtrusive program execution trace compression using branch predictor events. CASES 2010 Proceedings of the 2010 international conference on Compilers, Architectures and Synthesis for Embedded Systems, ISBN: 978-1-60558-903-9 (2010)
147. Vetro, A., Zazworka, N., Seaman, C., Shull, F.: Using the iso/iec 9126 product quality model to classify defects: A controlled experiment. In: 16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012), pp. 187–196 (2012). DOI 10.1049/ic.2012.0025
148. Villasana, G., Castello, R.: An agile software quality framework lacking. In: 2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014. Institute of Electrical and Electronics Engineers Inc. (2014). DOI 10.1109/WCCAIS.2014.6916549
149. Wagner, S., Goeb, A., Heinemann, L., Kläs, M., Lampasona, C., Lochmann, K., Mayr, A., Plösch, R., Seidl, A., Streit, J., Trendowicz, A.: Operationalised product quality models and assessment: The quamoco approach. Information and Software Technology **62**, 101–123 (2015). DOI 10.1016/j.infsof.2015.02.009
150. Wagner, S., Lochmann, K., Heinemann, L., as, M.K., Trendowicz, A., Plösch, R., Seidl, A., Goeb, A., Streit, J.: The quamoco product quality modelling and assessment approach. In: Proceedings of the 34th International Conference on Software Engineering, ICSE '12, pp. 1133–1142. IEEE Press, Piscataway, NJ, USA (2012)
151. Wagner, S., Lochmann, K., Winter, S., Deissenboeck, F., Juergens, E., Herrmannsdoerfer, M., Heinemann, L., Kläs, M., Trendowicz, A., Heidrich, J., Plösch, R., Goeb, A., Koerner, C., Schoder, K., Streit, J., Schubert, C.: The quamoco quality meta-model. [Online], October, 2012, [Accessed: 18.11.2017] (2012). URL https://mediatum.ub.tum.de/attfile/1110600/hd2/incoming/2012-Jul/517198.pdf
152. Xu, Z.: Fuzzy logic techniques for software reliability engineering. Ph.D. thesis (2001). AAI3001234
153. XXXXX, F.: 2nd International Conference on Network Computing and Information Security, NCIS 2012, *Communications in Computer and Information Science*, vol. 345. YYYY (2012)
154. Zadeh, L.: Fuzzy sets. Information and Control pp. 338–353 (1965)
155. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-ii. Information Sciences pp. 301–357 (1975)
156. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems **4.**(2.), 103–111 (1996)
157. Zadeh, L.A.: Is there a need for fuzzy logic? Tech. rep., Annual Meeting of the North American Fuzzy Information Processing Society (2008)
158. Zen Program Ltd: NDepend. [Online], [Accessed: 16.02.2018] (2017). URL https://www.ndepend.com/
159. Zhang, L., Li, L., Gao, H.: 2-D software quality model and case study in software flexibility research. In: Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA '08, pp. 1147–1152. IEEE Computer Society, Washington, DC, USA (2008). DOI 10.1109/CIMCA.2008.70

A Exact Queries in the Different Scientific Databases

Processing the amount of information in the returned publications required considerable effort. Consequently, the document search was repeated in the two most relevant computer science archives: ACM and IEEE, while preparing the manuscript extending the search dates till 04.10.2019 as reported in 3.2.

A.1 Q1a Query

Logical Query: ”Software Product Quality Model” OR ”Software Product Quality Framework”

Concrete Implementation in the Specific Databases:

1. ACM
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Date: 2000-2017
2. EBSCO
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Searching: Academic Search Premier DB
   Journal type: Peer-reviewed
   Date: 2000-2017
3. IEEE
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Date: 2000-2017
4. Science Direct
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Date: 2000-2017
5. Scopus
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Metadata: title, keyword, abstract
   Date: 2000-2017
6. Web of Science
   Search term: ”Software Product Quality Model” OR ”Software Product Quality Framework”
   Date: 2000-2017
   Databases: (1) Science Citation Index Expanded (SCI-EXPANDED) –1970-present, (2) Conference Proceedings Citation Index- Science (CPCI-S) –1990-present

B Q1b Query

Logical Query: (”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement)

Concrete Implementation in the Specific Databases:

1. ACM
   Search term: recordAbstract:(+(”execution tracing” logging) +quality +maintainability +software +(model framework))
   Date: 2000-2017
2. EBSCO
   Search term: (”execution tracing” OR logging) AND quality AND maintainability AND software AND (model OR framework)
   Searching: Academic Search Premier DB
   Journal type: Peer-reviewed
   Date: 2000-2017
3. IEEE
   Search term: ((”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement))
   Search Type: Abstract search
   Date: 2000-2017
4. Science Direct
   Search term: pub-date >1999 and title-abstr-key((”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement))
   Subject: Computer Science
   Date: 2000-2017
5. Scopus
   Search term: (”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement)
   Metadata: title, keyword, abstract
   Date: 2000-2017
6. Web of Science
   Search term: TS=((”quality model” OR ”quality framework”) AND assessment AND software AND analysis AND (measure OR measurement))
   Date: 2000-2017
   Databases: (1) Science Citation Index Expanded (SCI-EXPANDED) –1970-present, (2) Conference Proceedings Citation Index- Science (CPCI-S) –1990-present

C Raw List of Software Product Quality Model Publications Identified

The table below contains 125 sources identified in the scope of the systematic literature review for research question Q1. The duplicates are not listed. Each publication has been analysed. The decision, whether a publication is included or excluded in the review, is also documented with providing the reason. The list is alphabetically ordered by the domain descending, and by author and title in ascending order.

List of Abbreviations:

SPQM: Software Product Quality Model

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
1	Software Product Quality Model	yes	Not an SPQM	[20]
2	Software Product Quality Model	yes	Language Spanish	[26]
3	Software Product Quality Model	yes	Book, secondary source	[36]
4	Software Product Quality Model			[39]
5	Software Product Quality Model	yes	Process model with concepts from ISO9126	[53]
6	Software Product Quality Model	yes	Process model with concepts from ISO9126	[55]
7	Software Product Quality Model	yes	Not a model	[56]
8	Software Product Quality Model	yes	Book, secondary source	[63]
9	Software Product Quality Model			[66]
10	Software Product Quality Model			[77]
11	Software Product Quality Model	yes	Process model with some concepts taken over from ISO25010	[95]
12	Software Product Quality Model			[119]
13	Software Product Quality Model			[121]
14	Software Product Quality Model	yes	Process model	[127]
15	Software Product Quality Model	yes	Process model	[126]
16	Software Product Quality Model	yes	Process model with concepts from ISO9126	[136]
17	Software Product Quality Model	yes	Process model with concepts from ISO9126	[137]
18	Software Product Quality Model	yes	Book, secondary source	[144]
19	Software Product Quality Model			[147]
20	Software Product Quality Model	yes	Process model	[148]

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
21	Software Product Quality Model	yes	It is full conference proceedings volume with the abstract of the single papers' titles listed	[1]
22	Quality Model Assessment	yes	Safety standard not software product quality model	[112]
23	Quality Model Assessment	yes	Safety standard conformance, not software product quality model related	[113]
24	Quality Model Assessment	yes	Model for useability metrics not a full SPQM	[129]
25	Quality Model Assessment	yes	Not a software quality model	[5]
26	Quality Model Assessment	yes	Process model	[132]
27	Quality Model Assessment	yes	Information quality for software project management not software product quality	[15]
28	Quality Model Assessment	yes	Book, secondary source	[16]
29	Quality Model Assessment			[29]
30	Quality Model Assessment	yes	Software requirement specification not software product quality model	[89]
31	Quality Model Assessment			[27]
32	Quality Model Assessment	yes	Information quality frameworks not software product quality models	[37]
33	Quality Model Assessment			[38]
34	Quality Model Assessment			[44]
35	Quality Model Assessment	yes	Not software quality related	[52]
36	Quality Model Assessment	yes	Not a quality model but a description on the evolution of the quality models	[46]
37	Quality Model Assessment			[101]
38	Quality Model Assessment	yes	It is maintainability model not a complete SPQM	[23]

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
39	Quality Model Assessment	yes	Not computer scienece related	[90]
40	Quality Model Assessment			[91]
41	Quality Model Assessment			[14]
42	Quality Model Assessment			[159]
43	Quality Model Assessment	yes	It is a maintainability model, not a complete SPQM	[106]
44	Quality Model Assessment	yes	Not an SPQM	[34]
45	Quality Model Assessment	yes	Quality model for KPIs not for software product quality	[140]
46	Quality Model Assessment	yes	Not software product quality model related	[123]
47	Quality Model Assessment	yes	publication year: 1995	[54]
48	Quality Model Assessment	yes	It is not a software product quality model but a classification of the C++ source code modules in three risk groups: high, medium and low based on quality metrics. -publication year 1995	[143]
49	Quality Model Assessment			[150]
50	Quality Model Assessment	yes	Not an SPQM	[130]
51	Quality Model Assessment	yes	Not an SPQM	[80]
52	Quality Model Assessment			[62]
53	Quality Model Assessment	yes	The dissertation could not be obtained but newer research has been identified using similar methods.	[152]
54	Quality Model Assessment			[104]
55	Quality Model Assessment	yes	Not an SPQM	[108]
56	Quality Model Assessment	yes	Full conference proceeding	[153]

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
57	Quality Model Assessment	yes	Full conference proceeding	[2]
58	Quality Model Assessment	yes	Full conference proceeding	[3]
59	Manual Search			[67]
60	Manual Search			[65]
61	Manual Search			[7]
62	Manual Search	yes	It is a cost model	[8]
63	Manual Search			[68]
64	Manual Search			[17]
65	Manual Search	yes	Not an SPQM	[85]
66	Manual Search			[18]
67	Manual Search	yes	Not a complete SPQM	[9]
68	Manual Search	yes	Not a complete SPQM	[10]
69	Manual Search			[81]
70	Manual Search	yes	Process model	[21]
71	Manual Search			[43]
72	Manual Search	yes	Metric definitions, therefore they are excluded as separate metric definition are also excluded for other SPQMs.	[12]
74	Manual Search	yes	QM classification not a SPQM	[28]
75	Manual Search	yes	Safety standards related	[35]
76	Manual Search			[61]
77	Manual Search			[51]
78	Manual Search	yes	Not a complete SPQM, it describes maintainability only	[57]
79	Manual Search	yes	Not an SPQM	[58]
80	Manual Search			[59]
81	Manual Search			[64]
82	Manual Search	yes	Model for maintainability not a complete SPQM	[60]
83	Manual Search			[75]
84	Manual Search			[72]
85	Manual Search	yes	Not a model description but definition of metrics and measurement.	[71]
86	Manual Search	yes	Not a model description but definition of metrics and measurement.	[76]

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
87	Manual Search	yes	Not a model description but definition of metrics and measurement.	[69]
88	Manual Search	yes	Not a model description but definition of metrics and measurement.	[70]
89	Manual Search			[114]
90	Manual Search			[13]
91	Manual Search			[98]
92	Manual Search			[93]
93	Manual Search			[22]
94	Manual Search			[96]
95	Manual Search			[100]
96	Manual Search			[116]
97	Manual Search			[84]
98	Manual Search	yes	Not an SPQM.	[88]
99	Manual Search			[97]
100	Manual Search			[99]
101	Manual Search			[102]
102	Manual Search	yes	Not an SPQM.	[103]
103	Manual Search			[25]
104	Manual Search	yes	Not an SPQM.	[87]
105	Manual Search	yes	Not a model	[94]
106	Manual Search	yes	Comparison of models not a model description.	[24]
107	Manual Search			[111]
108	Manual Search	yes	Not a software product quality model but a model for enterprise architecture models	[117]
109	Manual Search	yes	Requirement traceability based on ISO/IEC 61508 to deal with safety. Not an SPQM.	[120]
110	Manual Search	yes	Book published in 1991	[122]
111	Manual Search			[32]

Table 8: List of Software Product Quality Model Publications Identified

No.	Domain	Excluded	Reason for Exclusion	Ref.
112	Manual Search			[141]
113	Manual Search	yes	Not a model description but a tool.	[142]
114	Manual Search			[50]
115	Manual Search			[30]
116	Manual Search			[110]
117	Manual Search			[124]
118	Manual Search			[134]
119	Manual Search	yes	Secondary source	[36]
120	Manual Search	yes	Not a model description but a tool.	[31]
121	Manual Search			[79]
123	Manual Search	yes	Not a model description but a tool. Text is in French.	[138]
124	Manual Search			[149]
125	Manual Search			[151]

D Evaluation of the Publications on Software Product Quality Models

The table below lists all of the 56 publications that define software product quality models or show tailoring of these models, based on the systematic literature review. The list is alphabetically ordered by (1) the domain descending, (2) model class ascending, and (3) relevance descending. Each identified publication is evaluated and a score of relevance is assigned.

Abbreviations:

Act. Actuality
Clar. Presentation Clarity
ETP Execution Tracing Present?
MS Manual Search
QMA The results produced by the query Q1b
Ref. Reference
Rel. Relevance
SPQM The results produced by the query Q1a

Table 9: List of Software Product Quality Model Publications Identified

Rel.	Act.	Clar.	Domain	Model Class	ETP	Model	Ref.
25	5	5	SPQM	ISO25010	no	Application of ISO25010	[66]
15	5	3	SPQM	ISO25010	no	Application of ISO25010	[119]
25	5	5	SPQM	ISO9126	no	Application of ISO9126	[121]
20	4	5	SPQM	ISO9126	no	ISO9126-application, research of ISO9126	[147]
15	3	5	SPQM	ISO9126	no	ISO9126 with clsutering and AHP for quality visualisation	[77]
0	4	0	SPQM	Metrics Framework for Mobile Apps	no	metrics framework	[39]
9	3	3	QMA	2D Model	no	2D Model	[159]
20	4	5	QMA	EMISQ	no	CQMM (process model for fitting EMISQ in the development process)	[91]
20	4	5	QMA	ISO25010	no	MDWE, ISO25010-adaptation, hybrid model	[29]
10	5	2	QMA	ISO25010	no	ASQuS , Application ISO-25010	[38]
12	4	3	QMA	ISO9126	no	ISO9126-implementation in model driven development	[62]

Table 9: List of Software Product Quality Model Publications Identified

Rel.	Act.	Clar.	Domain	Model Class	ETP	Model	Ref.
3	3	1	QMA	ISO9126	no	Fuzzy, Antiquarian Resource Digit Software, ISO9126	[104]
1	1	1	QMA	McCall et al.	no	QM for Java Agents with Classification	[14]
25	5	5	QMA	Quamoco	no	Quamoco	[44]
20	4	5	QMA	Quamoco	no	Quamoco	[150]
9	3	3	QMA	SQALE	no	SQALE	[101]
12	3	4	MS	ADEQUATE	no	ADEQUATE (prod, proc.)	[61]
6	2	3	MS	ADEQUATE	no	ADEQUATE (prod, proc.)	[79]
0	0	5	MS	Boehm et al.	no	Boehm et al.	[18]
15	3	5	MS	COQUALMO	no	Constructive QUALity Model (COQUALMO)	[110]
0	0	5	MS	COQUALMO	no	COnstructive QUALity Model (COQUALMO)	[17]
0	0	5	MS	Dromey	no	Dromey	[30]
9	3	3	MS	EMISQ	no	EMISQ	[124]
9	3	3	MS	EMISQ	no	CQMM (process model for fitting EMISQ in the development process)	[125]
10	2	5	MS	FURPS		FURPS+	[33]
0	0	5	MS	FURPS		FURPS	[49]
0	0	5	MS	FURPS	no	FURPS+	[48]
5	1	5	MS	GEQUAMO	no	GEQUAMO, ISO9126 derivate	[43]

Table 9: List of Software Product Quality Model Publications Identified

Rel.	Act.	Clar.	Domain	Model Class	ETP	Model	Ref.
0	0	5	MS	GQM	no	Goal Question Metric approach (hybrid model)	[134]
0	3	0	MS	IEEE Metrics Framework	no	IEEE Metrics Framework	[67]
20	4	5	MS	ISO25010	yes, in part.	ISO25010	[75]
15	5	3	MS	ISO25010	no	Application of ISO25010	[65]
15	3	5	MS	ISO9126	no	ISO9126 application	[22]
15	3	5	MS	ISO9126	no	ISO9126-application (tailored to commercial off-the-shelf (COTS) software product and software component evaluation)	[7]
10	2	5	MS	ISO9126	no	ISO9126-Fuzzy-AHP	[105]
5	1	5	MS	ISO9126	yes, in part.	ISO9126	[72]
9	3	3	MS	Kim and Lee	no	Kim and Lee	[81]
0	0	5	MS	McCall et al.	no	McCall et al.	[114]
5	1	5	MS	QMOOD	no	QMOOD	[13]
25	5	5	MS	Quamoco	no	Quamoco	[149]
20	4	5	MS	Quamoco	no	Quamoco	[151]
0	0	1	MS	SATC	no	SATC, hybrid	[64]
0	0	5	MS	SQAE	no	SQAE	[111]
10	2	5	MS	SQAE and ISO9126 combination	no	SQAE-ISO9126	[25]

Table 9: List of Software Product Quality Model Publications Identified

Rel.	Act.	Clar.	Domain	Model Class	ETP	Model	Ref.
25	5	5	MS	SQALE	no	SQALE	[99]
20	4	5	MS	SQALE	no	SQALE	[59]
15	3	5	MS	SQALE	no	SQALE	[100]
12	4	3	MS	SQALE	no	SQALE	[97]
12	4	3	MS	SQALE	no	SQALE	[102]
9	3	3	MS	SQALE	no	SQALE	[96]
5	5	1	MS	SQALE	no	SQALE	[98]
15	3	5	MS	SQUALE	yes, in part.	SQUALE	[11]
15	3	5	MS	SQUALE	no	SQUALE	[68]
3	3	1	MS	SQUALE	no	SQUALE	[116]
3	3	1	MS	SQUALE	no	Qualixo	[93]
0	0	5	MS	SQUID	no	SQUID	[84]

Table 9: Evaluation of the Publications on Software Product Quality Models