Information and Software Technology 70 (2016) 18–29

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Software reliability modeling based on ISO/IEC SQuaRE

Felipe Febrero, Coral Calero*, M. Ángeles Moraga

Instituto de Tecnologías y Sistemas de la Información, Universidad de Castilla-La Mancha, Paseo de la Universidad, 4-13071 Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 6 February 2015
Revised 29 September 2015
Accepted 30 September 2015
Available online 5 November 2015

Keywords:
Software quality
Software reliability modeling
International standard
SQuaRE

ABSTRACT

Context: The increasing dependence of our society on software driven systems has led Software Reliability to become a key factor as well as making it a highly active research area with hundreds of works being published every year. It would, however, appear that this activity is much more reduced as regards how to apply representative international standards on Product Quality to industrial environments, with just a few works on Standard Based software reliability modeling (SB-SRM). This is surprising given the relevance of such International Standards in industry.
Objective: To identify and analyze the existing works on the modeling of Software Reliability based on International Standards as the starting point for a reliability assessment proposal based on ISO/IEC-25000 "Software Product Quality Requirements and Evaluation" (SQuaRE) series.
Method: The work methodology is based on the guidelines provided in Evidence Based Software Engineering for Systematic Literature Reviews (SLR).
Results: A total of 1820 works were obtained as a result of the SLR search, more than 800 primary studies were selected after data filtering. After scrutiny, over thirty of those were thoroughly analyze, the results obtained show a very limited application of SB-SRM particularly to industrial environment.
Conclusion: Our analysis point to the complexity of the proposed models together with the difficulties involved in applying them to the management of engineering activities as a root cause to be considered for such limited application. The various stakeholder needs are also a point of paramount importance that should be better covered if the industrial applicability of the proposed models is to be increased.

1. Introduction

It is generally accepted that Reliability is a key factor in Software Quality since it quantifies failures and misbehavior. Also, on the economic point of view, high Reliability is desirable if the total costs of the software product are to be reduced. There are also other very important aspects such as customer dissatisfaction and loss of the manufacturer's prestige that can be traced to Software Product Reliability issues. It should not therefore currently be necessary to discuss the paramount importance of Software Reliability [1–4] in many sectors of industry and society since Software Reliability is the crucial factor as regards estimating both software quality and software cost [5]. According to Musa [6], "Reliability is probably the most important of the characteristics inherent in the concept Software Quality."

Developing performable ways in which to build reliable systems is therefore a real need, and knowing how to assess the actual reliability level of any software product is of no less importance. If this is to be achieved then it is necessary to develop models that are able to assess what level of reliability can be delivered by the software systems. This is the purpose of Software Reliability Modeling (SRM). However, if these models are to be effectively applied in day–to-day industrial practice then it is not only necessary for the proposed models to be sound, well founded and capable of being applied in an efficient, effective and economic manner, but also to be clearly valuable and profitable for the organization. International Standards play a central role in this issue since the objective demonstration of compliance with quality standards provides a means to demonstrate to client organizations that requirements are being achieved as well as generates better positioning in the market by means of increased customer satisfaction. Companies and organizations increasingly require their providers to comply with International Quality Standards.

Despite the above and the high research activity on Software Reliability there would appear to be very little activity in SRM based on Standards, which is surprising given the aforementioned relevance of International Standards in industry. In a previous work [7] we identified a potential lack of research as regards this point and the consequent need to conduct a Systematic Review on the basis of this result, along with the necessity to attain a better understanding of the applicability of SB-SRM to industry. This work is therefore the result of having searched for an answer to the question of how to apply

* Corresponding author. Tel.: +34 926295300; fax: +34 926295354.
E-mail addresses: ffebrero@computer.org (F. Febrero), Coral.Calero@uclm.es (C. Calero), MariaAngeles.Moraga@uclm.es (M. Ángeles Moraga).

http://dx.doi.org/10.1016/j.infsof.2015.09.006
0950-5849/

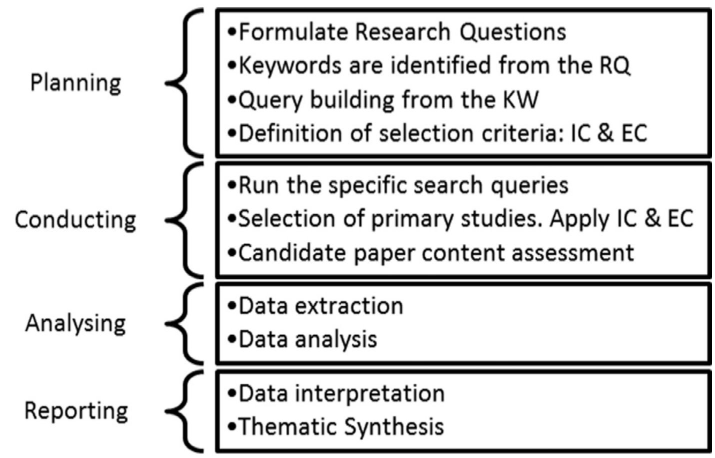

Fig. 1. Systematic Literature Review.

Software Product Reliability Modeling to industrial environments using representative international standards as a basis. To achieve this we first performed an SLR on SB-SRM in order to then integrate the results with the proposals from International Standards, all within the broader framework of Software Product Quality as a reference area in which the research is situated.

This paper provides two main contributions: firstly, it is the first systematic review of SB-SRM literature and secondly, it presents an innovative layout with which to model Software Reliability that integrates the needs of the different stakeholders in a simple but highly descriptive manner.

This paper is organized as follows. Section 2 presents the research work and the Systematic Review methodology as well as the framework chosen. Also in Section 2 Software Product Reliability is considered from the Quality Standards point of view. Section 3 shows the application of the SLR to the context of software reliability, along with the data extraction and results. Section 4 presents an analysis and a synthesis with the answers to our research questions. Section 5 presents a proposal to overcome the principal issues identified, as well as proof-of-concept which aim is shown the feasibility of the proposed approach. In Section 6 our main findings and threats to validity are discussed. Conclusions and further works are then summarized in Section 7.

2. Research description

2.1. Research methodology

As mentioned, this research is designed to follow the standard guidelines for SLR as specified in [8–12] and applied by previous SLRs in the Software Engineering area [13–15] from which valuable insight has also been obtained. A Systematic Literature Review is a process whose intention is to identify, extract and aggregate the best information from the available literature which, with the aim of mitigating bias, uses replicable methods to identify relevant studies and then to analyze those studies. This research methodology is outlined in Fig. 1.

The first stage comprises both a formulation of the problem and the establishment of a protocol that will drive the review. The objectives and the scope of the review are identified at this stage and are expressed by means of the Research Questions (RQ) from which keywords are derived. A Systematic Literature Review is driven by a very narrow research objective that is formalized by means of a short set of very specific research questions. It is also necessary to plan a search strategy by selecting which search sources will be used to find the primary studies. Inclusion Criteria (IC) and Exclusion Criteria (EC) are formalized in order to make it possible to include only primary studies that are relevant to answer the research question. These criteria must be straightforward to apply and, to mitigate each evaluator's bias, not require any interpretation.

The second stage is the data collection and evaluation process. This signifies searching for relevant papers that match the search string in each of the search sources selected. This is done by establishing the keywords and the particular search strings for each literature source. Once the search has been completed we can proceed to the Selection of the primary studies by applying the inclusion and exclusion criteria. The reviewers analyze the title and abstract in the search for terms and concepts that reflect the contribution of the paper. Once the selection phase has finished, the resulting works are analyzed to extract the data that is relevant to the research objectives. Finally, the overall process and outcomes are reported.

2.2. Research framework

This research falls into the knowledge area of Software Product Quality, and in this field the state-of-the-art is led by International Standard proposals such as those from ISO/IEC or IEEE and other like MIL-STD[1] or ECSS[2] for specific industrial environments. We have chosen the ISO/IEC 25000 "Software Product Quality Requirements and Evaluation" (SQuaRE) series of standards as a reference framework for this work. The rationale behind this selection is that the International Standards, and SQuaRE in particular, tackle the well-known lack of consensus and the variety of views on what Software Quality is bringing together the efforts of hundreds of volunteers representing varied viewpoints and interests but also SQuaRE is the most recent release on this field and then, arguably, offers the more mature proposal in the framework of Standard Based Software Product Quality.

In the SQuaRE proposal [16,17] the Quality of a system is understood as the degree to which the system satisfies the stated and implied needs of its various stakeholders. It is, thus, necessary to consider Quality from different stakeholder perspectives. The quality models provide a framework with which to collect stakeholder needs. Among the documents in the ISO/IEC standard the 25010 "Quality Model" defines a product quality model composed of eight characteristics which are further subdivided into sub-characteristics (Fig. 2). This model is understood as a structural model that SQuaRE defines as; "Quality model: defined set of characteristics, and of relationships

[1] United States defense standard, often called a military standard, "MIL-STD", "MIL-SPEC" or "MilSpecs".

[2] European Cooperation on Space Standardization.

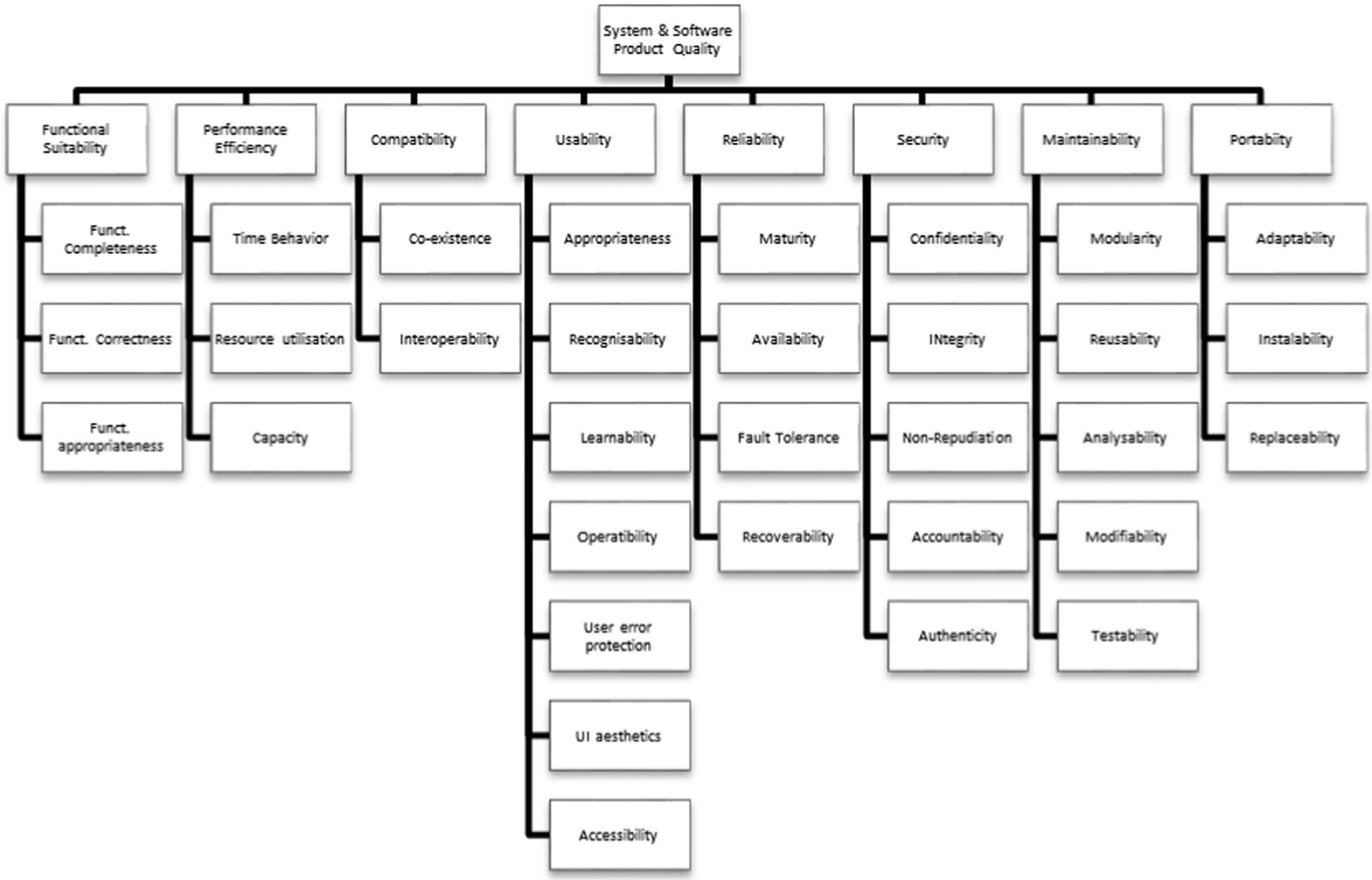

Fig. 2. SQuaRE quality model [17].

Table 1
SQuaRE reliability characteristic definitions.

Reliability	Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.
Availability	Degree to which a system, product or component is operational and accessible when required for use
Maturity	Degree to which a system meets needs for reliability under normal operation
Fault tolerance	Degree to which a system, product or component operates as intended despite the presence of hardware or software faults
Recoverability	Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system

between them.” Thus in this research modeling means analyzing the components and their relationships instead of reducing the observable phenomena into a set of mathematical formulae. The standard also provides definitions of both each quality characteristic and the sub-characteristics.

This Hierarchical Decomposition strategy is intended to provide a means to deal with the conceptual complexity in order to describe Quality as a multidimensional property, in addition to providing the means to obtain simple measurable attributes that are suitable for further combination into a quality index.

2.3. The research object

This research focuses on Software Product Reliability. Reliability is a broad concept that we apply whenever we expect something to behave in a certain way. Very different definitions for this concept have been proposed, from the classic hardware legacy definition in terms of probability of failure-free operation or the vision of Reliability as continuity of correct service to the more recent proposals that extend the vision of reliability in order to integrate the user’s perception of the system. This lack of consensus represents a major handicap and is a symptom of a topic still on development and somehow immature. The International Standards like those from IEEE and ISO/IEC organization represent a very valuable effort on the way to gain such a common agreement.

In the ISO/IEC SQuaRE [17] context Software Product Reliability is defined as the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. It is to be noted that this vision is broader than the classic one as probability of failure-free operation. Reliability is considered, then, as a combination of Availability, Maturity, Fault Tolerance and Recoverability concepts for which a formal definition, detailed in Table 1, is also provided although the nature of such a combination is not defined in the Standard.

The fact that the Standard does not discuss how these concepts relates one each to the others is a relevant hindrance to the applicability of the proposal as well as could be masking whether some similar concept is missing in the proposed description. It is our understanding, thus, that the relationship between those proposed components need to be analyzed and so we will do on Section 5.

It is also necessary to note that, reliability can and should be considered from different points of view in order to meet the different stakeholders’ needs. The SQuaRE proposal recognizes different stakeholders: Primary user, Secondary users and Indirect user as well as

Table 2
SQuaRE reliability views [17].

	USER			
	Primary user	Secondary users		Indirect user
		Content provider	Maintainer	
User needs	Interacting	Interacting	Maintaining	Using output
Reliability	How reliable does the system need to be when the user uses it to perform their task?	How reliable does updating the system with new content need to be?	How reliable does maintaining or porting the system need to be?	How reliable does the output from the system need to be?

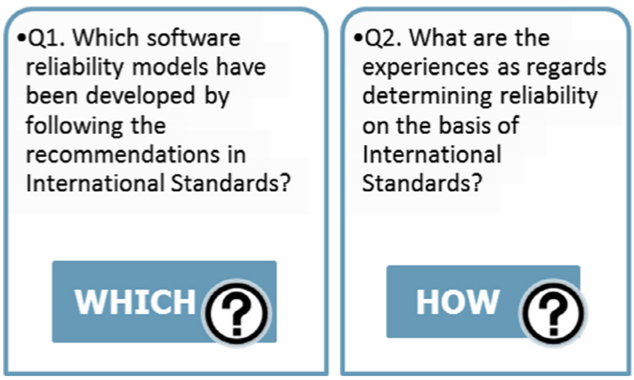

Fig. 3. Research questions.

their different needs, e.g. in the 25010 document [17] we can read (Table 2) the following:

Despite the fact that these are two extremely relevant points to consider the standard's documents provide little information on how to deal with them. Therefore, this research is very interested on investigate how both, academia and industry, have faced these issues; how the proposed sub-characteristics relates one each other and how the user's needs are captured, since they will greatly determine whether or not the proposal will result in an applicable schema.

3. The Literature Review

The present Systematic Literature Review involved three software engineering researchers, two of whom acted as primary screeners and the third of whom acted as an auditor. The Review was conducted by applying the process presented in Section 2. The search for primary studies was conducted in February 2014 to cover papers published between 1991 and 2014. This time span was chosen as a trade-off between feasibility and relevance, taking into account the release date of the ISO/IEC 9126, which is the antecessor of our chosen framework. In terms of the scope of the research, our interest lies in the software product, leaving aside the analysis of software production processes, and the variety of techniques used to build reliable software.

Our systematic review aims to, as far as possible, identify and analyze those primary studies that propose Software Product Reliability Models, theories or industrial experiences based on or closely related to International Standards. We have approached this aim by answering the following research questions (Fig. 3):

The first question Q1 aims to identify *which* relevant related work is available on the topic in which we are interested, i.e. quality or reliability models developed as a refinement of an International Standard proposal or adhering to its methodological recommendations or strategies. Q2 investigates *how* the proposed models have been or can be applied to either academic research or real industrial projects. This question also aims to investigate the difficulties involved in such an application. The ultimate goal is to obtain an insight into how to apply Software Product Reliability Modeling to industrial environments based on representative International Standards, paying particular attention to the impact of SQuaRE and ISO/IEC 9126 series.

The research questions and scope were used to derive a set of relevant keywords which has been further split into two sets related to domain specific concepts and generic contextual terms (Fig. 4).

As well as the inclusion and exclusion criteria, which are now formalized (Fig. 5) for further application.

The search was conducted using automated search engines of three of the largest and most complete scientific databases: IEEE Xplore www.ieeexplore.ieee.org, ACM Digital Library www.portal.acm.org and Science-Direct www.sciencedirect.com that were chosen because are widely recognized [10,12] as being an efficient means to conduct Systematic Reviews in the context of Software Engineering. We applied search strings based on the logic combination of the keywords identified above. As a result of this step, we obtained a total of 1820 papers: 423 from ACM digital library, 1161 from IEEE Xplore and 236 from Science Direct.

After the initial data preparation, the selection process comprised four iterations. The first was carried out separately by two reviewers while the following one was conducted by one evaluator. The inten-

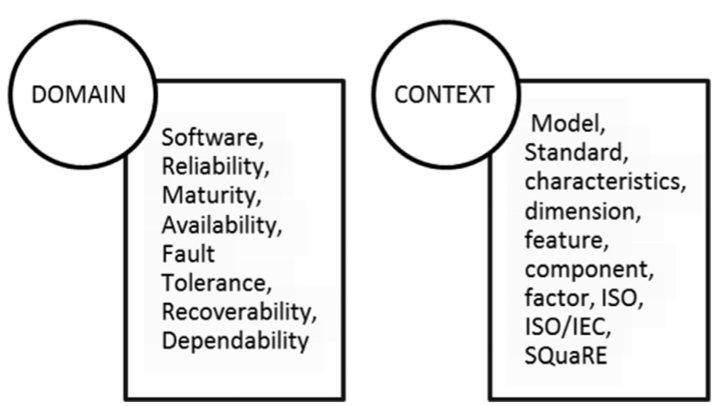

Fig. 4. Research keywords.

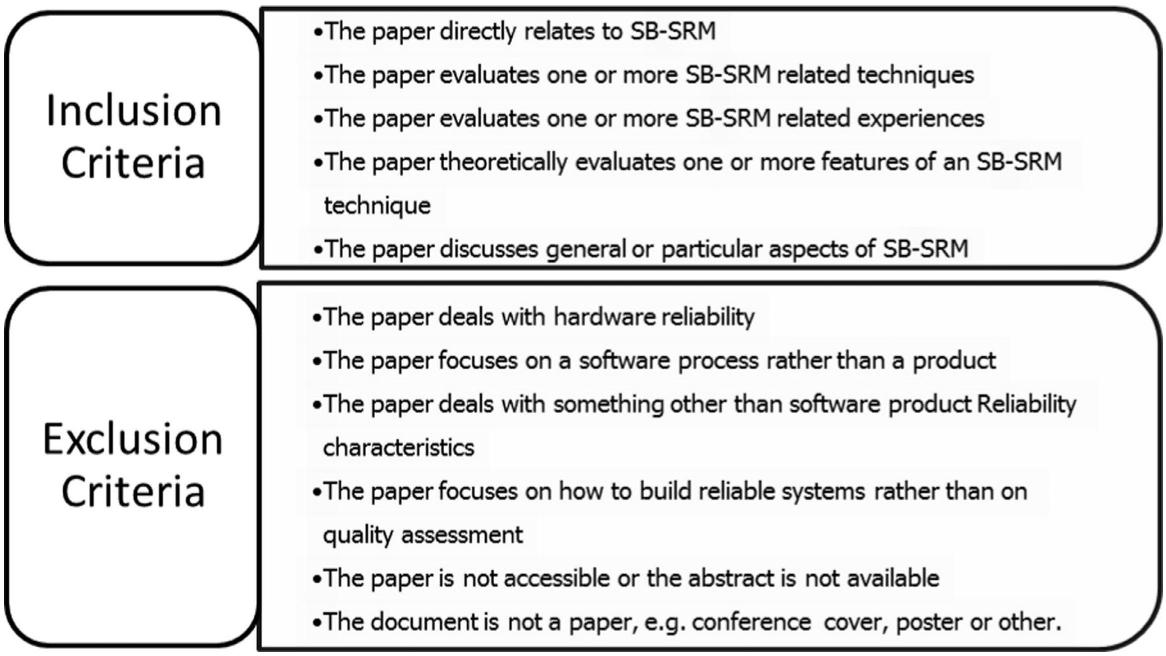

Fig. 5. Selection criteria.

Table 3
The selection process.

Review step	Rational	Works at output
Automated search	Results sought in DBs	1820
Data preparation	After duplicates rejection (1653)No paper, errors, hardware (1406)Not applicable fora (801)	801
Independent review	Review title & abstract, rejecting those out of scope or not relevant to our aim.	32
Auditing	Proposed papers review resolving discrepancies and rejecting those not relevant to our aim.	29
1st joint review	Paper's contents joint review	26
2nd joint review	Final decision on input to thematic synthesis	11

Table 4
Selected works.

	Work title	Inclusion criteria	Year	Reference
1	Software dependability evaluation based on quality characteristics	1,2	2010	[18]
2	Measurement of software requirements derived from system reliability requirements	1,4	2010	[19]
3	A comprehensive code-based quality model for embedded systems	1,2,3	2012	[20]
4	Using dependability benchmarks to support ISO/IEC SQuaRE	1,2,3	2011	[21]
5	Predicting quality of O.O. Systems using a QM based on design metrics & data mining	1,3	2009	[22]
6	Experience with the use of IEEE 982.1 standard in software programs	3	1997	[23]
7	Handbook-based high unit-value software reliability prediction method	4	2010	[24]
8	A process framework for customizing software quality models	1,2	2007	[25]
9	Analysis of contribution of conceptual model quality to software reliability	4,5	2010	[26]
10	Software quality and CASE tools	4,5	1999	[27]
11	Software reliability measurement. Use software reliability growth model in testing	4	2005	[28]

tion of this step was to reduce any bias in the results. The last two iterations were conducted as a joint review by the three researchers. Discrepancies regarding the inclusion and relevance of selected paper were discussed and agreed upon. Table 3 summarizes the figures of the selection process.

The first noticeable fact is possibly the important reduction in the number of papers from almost two thousand down to first thirty-two candidates which underwent an in-depth analysis. This is actually owing to the search strategy itself, since we performed a broad general search but were interested in a narrow research scope, hence the dramatic decrease in the amount of papers.

With regard to the results of the first review, the two independent researchers reached a high level of agreement. Of a total of 32 possible hits, 23 of these were selected by the two reviewers, 5 by only one and 4 by the other. A third researcher then reviewed this result with the purpose of audit selection and resolving discrepancies. A joint review agreed a first group of 26 works to be analyzed in depth. This review was conducted by a researcher and presented in a second joint review from which we obtained the 11 papers, on Table 4, that form the basis of the thematic synthesis. It is worth noting that nine of these were in the initial set of 23 papers chosen by both screeners. The classification had an agreement of 71.9%. The obtained kappa value was 0.85, which is a fairly good agreement.

3.1. Review results

In order to help us conduct this analysis in a systematic and structured manner, we chose an existing [29] classification of research approaches in addition to the research questions. First of all we analyzed what kind of research was performed. This data (Table 5) helped

Table 5
Research type and objective.

Work	Focus or stated objective of the paper	Research type
1	SQ assessment (reliability/dependability)	Proposal
2	Reliability requirements definition	Proposal
3	SQ assessment. (C++ source code).	Evaluation
4	System recoverability assessment	Proposal
5	SQ assessment	Proposal
6	Software reliability measurement program	Evaluation
7	S. reliability prediction/assessment	Proposal
8	S. Q. models application	Proposal
9	Factors influencing software reliability	Evaluation
10	Factors influencing software reliability	Evaluation
11	Software reliability measurement	Validation

us to understand what activities are being carried out in addition to providing a first insight into the impact of SB-SRM in academia and industry

It is interesting to note that almost all the evidence gathered falls into two kinds of research: “Proposal of a solution”, that is, works that propose a solution and argue for its relevance without a full-blown validation; and “Evaluation”, which corresponds to papers that investigate a problem in practice. Our own work is research consisting of the proposal of a solution as defined in the aforementioned reference.

Then we focused on what the selected works deal with. At a higher level we concentrated on whether the work is Reliability specific or it dealt with in a more general scope such as that of dependability [1,30] or Software Product Quality. The results in Table 6 show that when in the context of Standards, Software Reliability is often considered at a high level together with Quality or dependability. Regarding which particular Standards are being reported in the selected papers. Table 6 shows that the SQuaRE series still has a very limited impact since most of the available works reference its antecessor, ISO/IEC 9126. Others proposals like IEEE-std 982 [33] and IEEE-std 1633 [34] also have relevance in this area and are sometimes cited together with the ISO/IEC 9126.

In each paper we have also identified whether SQuaRE´s Reliability characteristics are treated on any way or not. We have also search for the Reliability characteristics considered in other relevant standards, e.g. Robustness as defined in the IEEE-Std 610.12 [31]. We have found that there is little analysis of sub-characteristics, and in most cases the papers do not in fact go beyond the first level of decomposition, associating measurable attributes directly with the high level characteristic.

Finally, we identify how the hierarchical decomposition and the measure aggregation are treated. The measurement process is only rarely explicitly detailed and the literature sometimes even refers to the values of attributes not explicitly defined in the document.

Only one of the works analyzed deals with the Operational Profile [32], which should play a central role in reliability assessment. It is also interesting to analyze whether the definition of reliability requirements is taken into consideration within the modeling frameworks, and it is worth noting that this is often not the case.

4. Evidence synthesis

The objective now is to build the answer to our research questions. In order to answer our, ‘which’ question we focused on which models or modifications to models are proposed, and which aspects related to Reliability factors are dealt with. We answered our ‘how’ question by searching for reports on models applied to real industrial projects or academic experiments and analyzing how the concept of reliability is treated and understood when in the SB-SRM context, in addition to the specific techniques that are applied.

4.1. *Answering the which*

The most common approach is that of taking the Standard’s proposal as a starting point for a model proposal. Such as in [24] where the author states that the proposed model is constructed on the basis of both ISO 9126 and other similar models like those of McCall, Boehm or FURPS. However, no explanation is provided as to how it is developed beyond adherence to the strategy of hierarchical decomposition. The Proposed model is further empirically analyzed in the context of the Object Oriented paradigm.

In [18] the authors present a method for evaluating reliability/dependability (the two terms are used interchangeably) with reference to the quality model of ISO/IEC 9126. The use of the standard is limited to guiding the selection of the quality index. They do not propose measures. However, a measure aggregation proposal is presented by combining weights with the use of techniques from Fuzzy Theory and the Analytical Hierarchical Process.

In [21] the authors address the problem of integrating COTS software into the product and the need to ensure global quality, focusing on Recoverability, a sub-feature of Reliability, and how SQuaRE deals with this problem. These authors propose enriching the ISO/IEC 25045, thus by extension SQuaRE, integrating the concept and techniques of dependability benchmarking in order to tackle the limitations in the evaluation of software products in the presence of generic

Table 6
Research topics.

		Study										
		1	2	3	4	5	6	7	8	9	10	11
Scope	Quality			•		•			•		•	
	Dependability	•			•							
	Reliability		•				•	•		•		•
Framework	SQuaRE			•	•					•		
	ISO-9126	•	•			•					•	•
	Others		•				•	•	•			
Characteristics	Maturity		•							•		•
	Availability									•		
	Recoverability		•		•					•		•
	Fault Tolerance		•							•		•
	Robustness				•							
Techniques	Decomposition					•						
	Operational profile				•							
	Aggregation	•		•	•							
	Measures			•	•		•			•		
	Requirements		•	•								

Table 7
Research question's answer.

	Which	How
1	A model proposal including measure aggregation method	Taking the standard as starting point and applying Fuzzy Theory & AHP
2	Conceptual analysis of reliability	By means of an operationalization strategy based on treating *R* as a functional requirement.
3	A model proposal	Hierarchical structure integrating ISO/IEC 25010
4	SQuaRE model completeness analysis	Enriching/extending the proposal
5	Reliability assessment	Estimating missing data from I.S
6	Report of application	Measure selection guided by I. Standard
7	A model proposal	Taking the standard as starting point
8	Capturing stakeholder's expectations	Converting the quality model into a survey questionnaire
9	Impact of factors on delivered Reliability	By means of a survey study
10	Impact of factors on delivered Reliability	By means of a survey study
11	Capturing stakeholder's expectations	By means of a survey questionnaire based on ISO/IEC 9126

disturbances. In [22], the conventional determination of Reliability is shown in addition to how parameters can be applied on the basis of standard based models to estimate participation in computing the Reliability of missing data such as the initial design fault density.

Several works tackle the conceptual analysis of software reliability and related concepts. As mentioned in Section 2, one of the more controversial issues is the definition of Software Reliability itself and the variety of related concepts proposed in the different standards. In [19] the authors analyze the variety of reliability views and propose to harmonize the treatment in different standards (ECSS, ISO9126 and IEEE-stds) into a generic standard-based reference model in which reliability is expressed as a set of functional requirements and is therefore implementable and measurable. This is intended to ease the application of reliability engineering and also contributes to filling the gap between the upper abstraction levels in the conceptual decomposition and the technical attributes at lower levels.

There is some available evidence regarding the impact that a variety of factors have on the Reliability delivered. The influence of the production process has been analyzed in several of the selected works, usually by means of survey studies. In [27] the authors examine the impact of both back-end and integrated CASE tools on the quality, and hence the reliability, of the software product using the ISO/IEC 9126 quality definition to drive the survey research. In [26] the guidance of ISO/IEC 25010 is used to analyze how the quality of conceptual models influences software reliability and which factors of conceptual model quality have an effect on software reliability.

4.2. Answering the how

A first finding when considering how to apply SB-SRM is the need to tailor the model to the particular needs in a case by case manner despite what the standards do not provide details on how to proceed and most of the analyzed works merely show the final tailored model. However, survey research is reported in [25] that seek to develop and validate a new process framework with which to customize software quality models in order to meet organizational goals which is a requirement for the application of generic quality models. The author proposes to convert the software quality model into a survey questionnaire as it is also proposed in [27]. Jung and Yang [28] propose using a survey based on ISO/IEC 9126 reliability metrics, later processed following the Analytical Hierarchical Process, to request information regarding customer satisfaction in order to guarantee that stakeholders' expectations are met.

Another basic task to be dealt with is the selection of measures. SQuaRE does not provide a specific method to guide this activity. However, Farr [23] reports the application of the IEEE 982.1 standard to two industrial projects. The role of the standard was to provide guidance for the measure selection process.

In [20] the authors report the development of a quality model oriented to the internal quality of embedded source code which covers requirements for the source code. The model is hierarchical in nature and integrates the vision proposed in the 25010 standard. The rational to dismiss the classical strategy of hierarchical decomposition and instead of refining the quality characteristics selecting first the low level source code quality features to then integrate the quality framework are the difficulties to make the link between the very high level nature of the proposed model on the standards and the source code properties and so with the implementation. It is worth to note that such is the problem that [19] intend to address from a different perspective based on the operationalization of the high level quality requisites. As mentioned in most cases the papers do not go beyond the first level of decomposition, associating measurable attributes directly with the high level characteristic.

The aggregation strategy for the low level attributes is a central issue when using hierarchical decomposition as a basis for modeling, despite which it is only occasionally dealt with in literature. In many cases the proposed aggregation of measures consists of a more or less sophisticated weighted sum. In [18] the researchers present a measure aggregation strategy based on the combination of subjective and objective weights to determine the final weight. The need to define and use an operational profile for the determination of reliability is well known, but only one of the selected papers [21] mentions this. This is almost certainly because the operational profile is more closely related to both the fault process and the testing activities.

A summary of the main subject of each of the analyzed works, in terms of our 'which' and 'how' research questions, is presented in Table 7. It is to note that some of the studies deal, also, with issues out of the scope of our research. Those issues are not detailed on our synthesis.

4.3. Hindrances

Finally, since one of our objectives is to analyze the reasons for the at least apparently limited application of SB-SRM, we searched the selected works for reported problems and hindrances to the industrial development of these programs. The outcome of this is that the literature reviewed reports on several issues that particularly hamper the effective application of the models analyzed. The complexity of the Reliability concept, which is characterized by its multiples facets, is shown in the majority of them. Work needs to be done to afford this in the simplest manner in order to enable industry to apply it. In [18] it is pointed out that: the comprehensive evaluation of the software is a complex issue, it is difficult to carry out proper evaluations and different methods have many different results. Linked to this complexity is the variety of definitions and factors analyzed in [19] that greatly hampers the software provider as regards meeting customer expectations.

Another recurrent problem [20,22] concerns the difficulties involved in relating the low level of measurable attributes of the source code or the design documents with the high level characteristics. Failing to make this link in a sound, understandable and simple manner implies that the models are not greatly effective in supporting the

Table 8
Hindrances to SB-SRM application.

Issue	Approach
Conceptual complexity & variety of definitions	Adherence to International Standard proposal
Failing to make the link between the very high nature of the proposed model and the source code properties in a sound understandable and simple manner	Hierarchy layout refurbishment
Failing to capture stakeholders expectations	Reliability as user oriented concept considering the variety of stakeholders viewpoints
Possible lack of completeness of the model	Explicit treatment of Robustness an Stability
Lack of sound and founded aggregation strategies	Treatment as decision problem (higher levels) Treatment as analytical problem (lower levels)

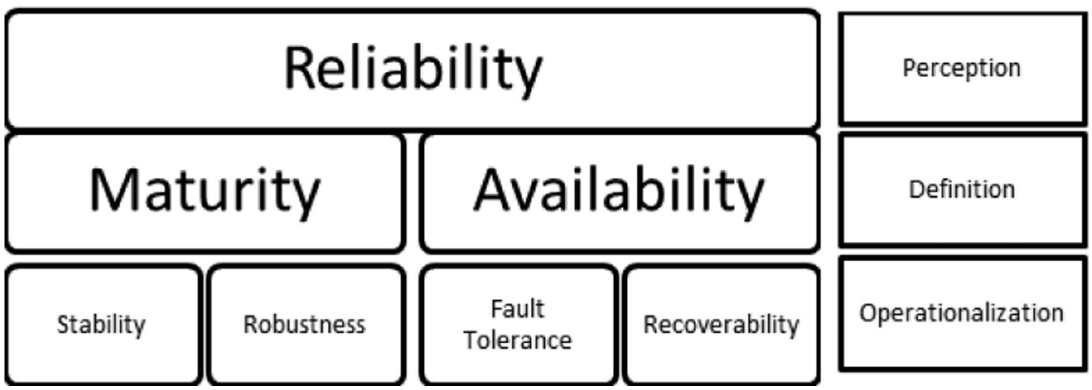

Fig. 6. Reliability model layout.

decision making process that the industry's stakeholder needs, and are not therefore used. Related to this is the apparent lack of aggregation strategies for the proposed models. Also, some works [21] point to a possible lack of completeness of the models. All of the above probably lies at the root of the lack of commitment mentioned in [23] and industry needs clear models and methods that are straightforward to apply.

On the Table 8 we have summarized the main identified obstacles for a broad application of SB-SRM together with a selected approach, that we will develop along the next Section 5, to overcome them.

5. The proposal

As discussed in Section 2, Reliability is a user-oriented concept. In the simplest sense, reliability is an assessment of how well system users think it provides the service they require. It is also important to consider that different users will have different viewpoints. Our proposal addresses this issue by considering the nature of the Reliability characteristics and its relation to different users' needs. The Reliability model therefore provides a framework with which to collect stakeholder needs.

As mentioned in Section 1 professor Musa claims [6] that Software Reliability concerns itself with how well the software meet the requirements of the customer, also in the seminal work on dependability [30] we read that Reliability is the continuity of a correct service. That is, a system will be reliable to the extent that it is available whenever it is required for use and behaves as expected by its users. Adhering to ISO/IEC vocabulary Reliability is, therefore, a certain function of Availability and Maturity in which maturity summarizes the correctness of behavior that is consistent with stakeholders´ expectations.

Maturity means then that user expectations are met and thus no changes or corrections are consequently required, which is Stability as understood on IEEE 982.1 when defining the Software Maturity Index. Fulfilling user expectations also imply a system that permits an easy and flexible use of the facilities, which can be mapped to Robustness in the terms of IEEE 610.12. This is directly related to the correctness of implementation since Robustness is understood [31] as the degree to which a system or component can function correctly in the presence of invalid inputs or a stressful environment. In summary, Maturity is made up of Stability and Robustness. On the other hand, the Availability of a system depends on how much it fails as regards the effort required to repair it. Availability is thus a certain function of a system´s Fault Tolerance, which determines whether a fault will manifest itself as a failure and its Recoverability, which accounts for the recovery efforts after failure.

The above discussion has been used to derive a new schema, see Fig. 6, for the decomposition of Reliability that can be mapped onto different stakeholders' needs, thus addressing the issue of capturing different viewpoints and needs.

Each level in this proposed hierarchy deals with both a particular description level and a particular user need. At the top we have Reliability as the user's perception of the system behavior, this global view being characteristic of end user or higher management levels. Below, the first decomposition levels enable the functional analysis of the global contributors to the user´s perception. The system can be analyzed at this level so as to make decisions regarding the definition of the quality characteristics by considering external constraints and business objectives in a simple manner, i.e. the maturity in terms of updates to the product can be largely impacted by the business model or the recovery delays by extrinsic constrains on operations safety. It is still necessary to operationalize the description to lower levels so as to be able to implement corrective actions or produce measures that can be used as input to compute high level indexes. The first step in that sense is the third decomposition level where still on the basis of Reliability characteristics proposed on International Standards the description is specialized onto secondary characteristics. This level enables us to obtain a quantitative estimation of the system behavior, thus external metrics according to ISO/IEC vocabulary. We could, e.g. estimate the Stability by means of the Maturity Index as per IEEE 982.1 [33] proposal or the Robustness using the Breakdown Avoidance while Fault Tolerance can be estimated by means of the Mean Time Between Failure and Recoverability by Mean Recovery Time as per ISO/IEC 9126 definitions. Bearing in mind that the

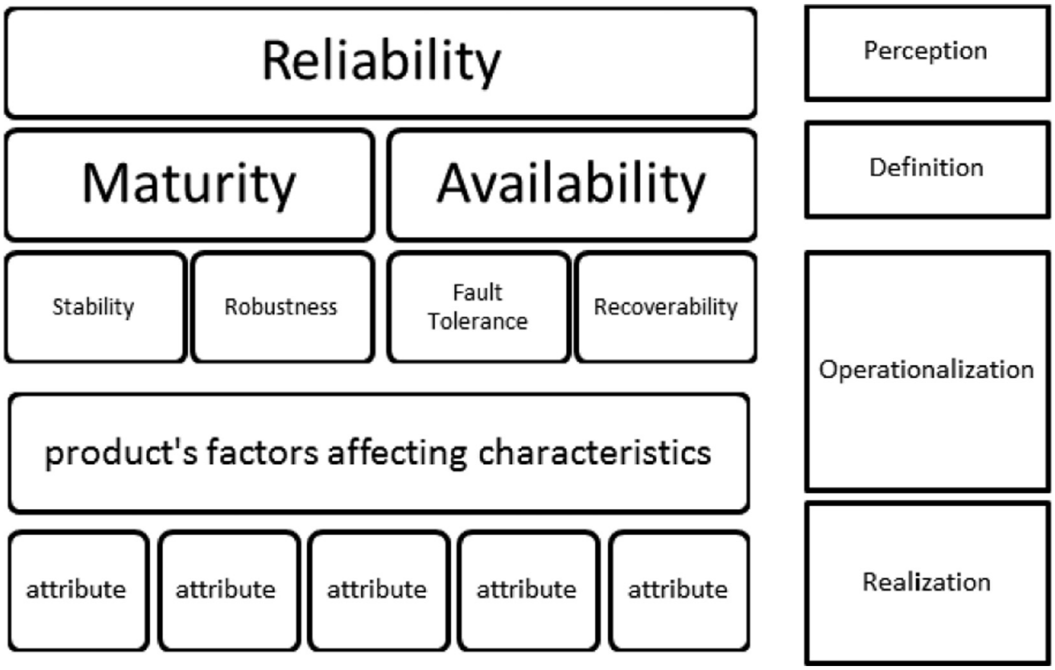

Fig. 7. Reliability assessment schema.

fundamental reason for this modeling is organizing knowledge to support business decisions, we suggest that treating it as a decision making model is the more appropriate, and thus the appropriateness of aggregation techniques like the Analytical Hierarchical Process.

Notwithstanding, the above still do not correspond to those low level software product attributes which are directly observable and measurable, let say lines of code or Function Points, but to system behavioral assessment. As a consequence, and also on the basis of the discussion presented in Section 4, our proposal introduces another level below the hierarchy defined on the basis of the ISO/IEC characteristics and sub-characteristics. The use of this description level is required to maintain the link between the user perception and managerial needs and the engineering levels in charge of the product implementation addressing another of the most important issues identified on our SLR, the difficulties to establish the relationship between the high level concepts on models definition and the product implementation. This additional level is, thus, intended to provide the gateway between such Software Product's observable attributes and the high level model's characteristics so concerned with properties and operation factors such as software complexity, specification changing rate, operational environment or test coverage which, despite being of a lower abstraction level than the model characteristic, are still above the measurable attributes to with they are related.

The integration of this low level component allows us to obtain a complete description, from the user's perception to the static measurable attribute and that mapping to the different user needs. This simple layout is still a powerful description of the multiple facets of Software Reliability, in addition to providing the means to resolve the main issues extracted from the literature review: the complexity of the descriptions and the difficulties involved in applying them as a decision tool in control and management activities. It also makes explicit how the different stakeholder needs can be covered in an integrated schema, as is depicted in Fig. 7.

5.1. Proof of concept

In order to illustrate how the proposal can be easily developed in practice we have performed the following proof-of-concept. The example consists of applying, as measure aggregation strategy, a recognized and widely used decision making approach, the Analytical Hierarchy Process (AHP) [35,36] which is a well-known decision theory and a technique for computing priorities. As inputs to the assessment we will use both, expert judgments and analytical objective measures. The AHP is composed (see Fig. 8) of the following main steps:

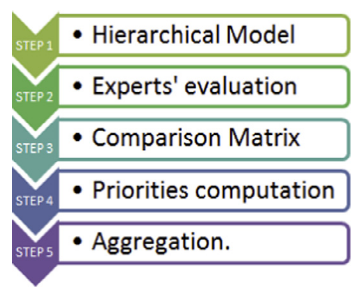

Fig. 8. Steps of the AHP.

Step 1: First, the problem has to be modeled into a hierarchy e.g. our proposed schema, which is fundamental to the process of the AHP. Hierarchy indicates a relationship between elements of one level with those of the level immediately below. In our case, we have the hierarchy shown in Fig. 6.

Step 2: Once the hierarchical model has been built data are collected from experts or the concerned users. The objective is comparing the hierarchy elements to one another with respect to the impact on their parent node in the hierarchy. This is achieved by means of the expert panel's answers to questions of the general form, "How relevant is element A as regards to element B?" known as "pairwise comparison". Answers can be provided in qualitative form or in numerical ratio scale. In order to apply this step, we have requested to an experts panel to perform the pairwise comparison regarding the relative relevance of each of the reliability's sub-characteristics on our proposed layout. This panel being composed by three experimented (about fifteen years of professional practice) software engineers

Table 9 sum-up the assessment of the experts, where *X*/*Y* means that the characteristic on the row is *X*/*Y* times more important than

Table 9
Pairwise comparison.

	Maturity	Availability	F. tolerance	Recoverability	Robustness	Stability
Maturity	1	13/9	5/2	5/2	17/12	13/3
Availability		1	7/3	7/3	11/6	7/5
F. tolerance			1	19/12	7/3	9/5
Recoverability				1	7/3	4/5
Robustness					1	29/15
Stability						1

Table 10
Computed priorities.

Local Priorities		Global Priorities
$w_M = 13/22$	$w_{St} = 15/44$	$W_{St} = 0.2015$
	$w_{Rb} = 29/44$	$W_{Rb} = 0.3984$
$w_A = 9/22$	$w_{Ft} = 19/31$	$W_{Ft} = 0.2507$
	$w_{Rc} = 12/31$	$W_{Rc} = 0.1584$

the one on the row i.e Availability is 7/3 more important than Recoverability, and conversely Recoverability is 3/7 more important than Availability.

Step 3: These pairwise comparisons gathered at previous step are now arranged into the comparison matrix at each level and according to the hierarchy structure. A comparison matrix is a square matrix $A = [a_{ij}]$ where the element a_{ij} is the relative importance of element i compared to element j. The diagonal elements of the matrix are 1 and a_{ji} can be assumed to be $1/a_{ij}$.

Step 4: Then, the vector representing the relative weights, or local priorities, of each of the attributes, can be found by computing the normalized eigenvector corresponding to the maximum eigenvalue of each matrix. This vector can be computed by dividing each element of the matrix with the sum of its column, then averaging the values in each row. Table 10 presents these local priorities on the left side, where w_x accounts for the priority or relative relevance of each characteristic regarding the one on their parent node e.g w_M accounts of the local priority for Maturity and w_{Ft} for Fault Tolerance.

Step 5: To aggregate the priorities along the hierarchy we have to consider that local priorities are calculated among same level elements and global ones are the product of all local weights from leaf to root on the hierarchy. In other words, the local priorities correspond to the relative weights of the nodes within a group of siblings with respect to their parent. Global priorities are obtained by multiplying the local priorities of the siblings by their parent's global priority i.e. the global priority for Stability is W_{St} computed as the product of w_{st} and w_M which are the local priorities on the path from the leaves to the root of the hierarchy. Table 10 sums-up this computation.

In that simple, but recognized and well founded way, we are capturing the view on Reliability for the particular group of stakeholders for which this expert's panel is representative. It is also easy to present this view in a graphical way without information loss e.g. by means of a Kiviat's diagram, what make it accessible to non-experts users as we can see on Fig. 9, where ST means Stability, RB Robustness, FT Fault Tolerance and RC Recoverability.

At that point, and to produce a reliability assessment showing how this method provides a simple way to incorporate experts' subjective values together with factual data we have requested to the same panel their subjective evaluation E_{FT} & E_{RB} (have to be normalized) on the Fault Tolerance and Robustness Reliability characteristics for a particular very-large distributed real-time system on which all of them are involved from more than ten years. We have, also, defined two analytical metrics; σ for the system stability based on the Maturity Index proposed on IEEE 982.1–1988 [33] and λ for the Recoverability re-using the normalized Mean Recovery Time (MRT) as proposed on ISO/IEC 9126. Table 11 sums-up the computation of the Reliability index.

Fig. 9. Reliability view representation.

Table 11
Reliability computation.

Inputs definition		
$\sigma = (m_t - \Delta m)/m_t$	$\lambda = 1 - T_M/T$	Expert's estimations
m_t : total number of modules	T_M: MRT	$E_{FT} = 13/9$
Δm: number of modified modules	T: total time	$E_{RB} = 61/18$
Index computation		
$R = W_{St}\,\sigma + W_{Rb}\,E_{RB} + W_{Ft}\,E_{FT} + W_{Rc}\,\lambda$		
$R = 0.1612 + 0.27 + 0.0724 + 0.1108 = 0.6144$		

Finally, such outcome could also be presented as a Kiviat's diagram, as in Fig. 10, which allows a joint description of the ideal objective and the actual assessment which is greatly helpful for the decision makers.

In the presented case it is clear that the estimated Fault Tolerance is far from stakeholder's expectations while the rest of characteristics are rather close to user's requirements. Further details on the practical application of this method can be found on [37].

6. Discussion

There is a need to assess reliability, and to be able do so the industry requires models that easily and effectively capture the complexity of this multifaceted concept. In order to obtain such models that can be applied in daily practice they must not only be sufficiently descriptive and simple to apply but must also be able to show that they are profitable to the organization.

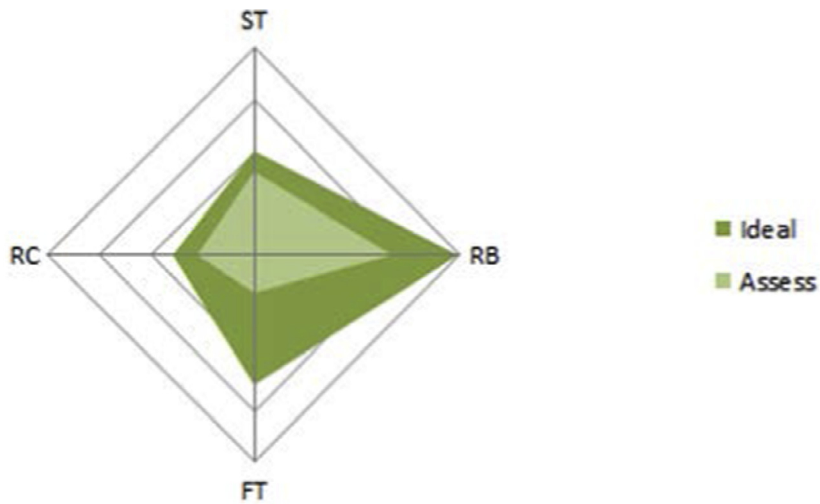

Fig. 10. Reliability assessment.

The evidence collected in this review shows that the main disadvantage of software reliability models is that they are too complex to be effectively applied in a daily practice. At the root of this high complexity is not only the very nature of Software Reliability itself but also the immaturity of a topic in which the first controversial issue is the variety of definitions the available literature offers for the study object. We have, also, observed that the terms Reliability and dependability are often used interchangeably and once even as synonyms. This lack of consensus represents a major handicap to the development of effective models. The international standards can play a first order role in order to gain such a consensus and thus ease the industrial application of software reliability modeling, not only by offering guidance for the modeling task itself, but also because they are recognized by industry as a trustable source of recommendation and guidelines.

Another important issue that is not always taken into account is that Reliability is also a perception (maybe mainly a perception) on how the system behaves. The different stakeholder needs is thus a point of paramount importance that should be better covered in order to increase the industrial applicability of the proposed models. Our proposed schema allows capturing the different views in a fairly simple manner, addressing the main difficulties involved in its application in industrial environments.

The upper levels in the presented layout facilitates project managers and the practitioners to assess of a software system based on sound and well-known behavioral assessment indexes which is appropriate to the needs of those stakeholders. The drawback is that such behavioral indexes do not offer detailed enough guidance as to easily acting on the software product in order to take corrective actions when required. We conclude thus the need of an additional level that of factors impacting software quality, intended to make the link between the quality characteristics and the measurable attributes.

6.1. Threats to validity

Despite the fact that this study has been carried out by following the SLR methodology, there may be some threats to its validity. The principal limitations concern the limited access to sources, a circumstance that may have led to a bias in the selection of publications owing to the possible existence of studies of interest in other databases. However, the databases used cover the area of software engineering well [10,12] and we have no reason to believe that this does not apply to software reliability. We are, therefore, reasonably confident that we are unlikely to have missed many significant published studies. Some relevant papers might not have been found in the digital databases when using our search and selection protocol. Automated searches rely on both search engine quality and how researchers write their abstracts. Although we are reasonably confident as regards how well digital databases classify and search indexed work, if abstracts and keywords are of poor quality it is clear that the search will be greatly flawed. However, since our selected keywords are commonly used terms the possibility of any significant contribution not mentioning these key words in the title or abstract is minimal. Another threat that needs to be considered is the possibility of bias in selection. This concern was addressed as described in Section 3, the selection phase was conducted independently by two authors then audited by the third one and finally reviewed in a joint revision. Finally, only studies published in the English language were selected in the search although the eventual bias owing to this have to be minimal since English is the most widely adopted language as regards writing scientific papers.

7. Conclusion and future work

This paper presents the results of an SLR on software reliability assessment based on representative International Standard proposals. The main outcome of this work is the confirmation that SB-SRM is receiving limited attention from the academic community in addition to having little impact on industry, or at least industry is not reporting on its application. Some clues can be obtained from the literature reviewed. The complexity of the concept itself is possibly the main impediment to the broad application of Software Reliability models in an industrial environment, but also that it is necessary to consider reliability from different perspectives in order to meet the different stakeholders' needs. This point seems to be receiving very little attention. Work needs to be done to afford this complexity in the simplest way in order to enable industry to apply it. The efforts of the different Standardization Organization are good examples of this, although they do not appear to have had a great impact on academia.

As future work we suggest analyzing the application of different aggregation strategies depending on the description level in the model. We advance the appropriateness of decision-making analysis for the upper abstraction level while common analytical techniques are more suitable for application at lower levels that are in accordance with the profile of the stakeholder concerned. We are also interested on the application of Bayesian Networks as alternative to the decision making techniques and performing a comparative analysis of both strategies. In both cases it is our aim to analyze how to develop a hierarchy concerning the variety of factors affecting the reliability sub-characteristics, we will do that on the basis of existing taxonomies like e.g. the one proposed by Beizer [38].

Acknowledgements

This work has been funded by the GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01) and by the VILMA project (Consejería de Educación, Ciencia y Cultura de la Junta de Comunidades de Castilla La Mancha, y Fondo Europeo de Desarrollo Regional FEDER, PEII-2014-048-P).

References

[1] B. Littlewood, L. Strigini, Software reliability and dependability: A roadmap, in: A. Finkelstein (Ed.), The Future of Software Engineering, ACM Press, 2000, pp. 177–188.

[2] M.R. Lyu, Handbook of Software Reliability Engineering, IEEE Computer Society Press and McGraw-Hill, 1996, pp. 3–9.

[3] B. Littlewood, Theories of software reliability: How good are they and how can they be improved? IEEE Trans. Softw. Eng SE-6 (5) (1980) 489–500.

[4] H. Pham, M. Pham, Software Reliability Models for Critical Applications, Idaho National Engineering Laboratory, 1991 Report EG&G-2663.

[5] W. Farr, Software reliability modeling survey, Handbook of Software Reliability Engineering, IEEE Computer Society Press and McGraw-Hill, USA, 1996, pp. 71–117.

[6] J.D. Musa, W. Everett, Software-Reliability Engineering: Technology for the 1990 s., IEEE Software, 1990, pp. 36–43.

[7] F. Febrero, C. Calero, M.A. Moraga, A systematic mapping study of software reliability modeling, Inf. Softw. Technol. 56 (2014) 839–849.

[8] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Keele University and Durham University Joint Report, 2007 Technical Report EBSE 2007-001.

[9] T. Dyba, B. Kitchenham, M. Jorgensen, Evidence-based software engineering for practitioners, IEEE Softw. 22 (1) (2005) 58–65.

[10] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. Softw 80 (4) (2007) 571–583.

[11] T. Dyba, T. Dingsoyr, G.K. Hanssen, Applying systematic reviews to diverse study types: an experience report, First Int. Symp. Empir. Softw. Eng. Meas. (2007) 225–234.

[12] B. Kitchenham, D. Budgen, P. Brereton, Using mapping studies as the basis for further research – a participant observer case studies, Inf. Softw. Technol. 53 (2011) 638–651.

[13] R. Wendler, The maturity of maturity model research: A systematic mapping study, Inf. Softw. Technol. 54 (2012) 1317–1339.

[14] D. Budgen, A.J. Burn, O.P. Brereton, B.A. Kitchenham, R. Pretorius, Empirical evidence about the UML: a systematic literature review, Softw. Pract. Exp. 41 (4) (2011) 363–392.

[15] C. Calero, M.F. Bertoa, M. Moraga, A systematic literature review for software sustainability measures, in: Proceedings of the 2nd International Workshop on Green and Sustainable Software GREENS, 2013, pp. 46–53.

[16] ISO/IEC 25000, Systems and software engineering – Software product Quality Requirements and Evaluation SQuaRE. 2005.

[17] ISO/IEC 25010, Systems and software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Software product quality and system quality in use models. ISO 2010.

[18] L. Xiaoli, L. Chao, D. Wenrui, W. Lei, Software dependability evaluation based on quality characteristics, in: Proceedings of the the 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 3, 2010, pp. 113–117.

[19] K.T. Al-Sarayreh, A. Abran, L. Santillo, Measurement of software requirements derived from system reliability requirements, in: Proceedings of the Workshop on Advances in Functional Size Measurement and Effort Estimation ACM, 2010, pp. 1–6.

[20] A. Mayr, R. Plosch, M. Klas, C. Lampasona, M. Saft, A comprehensive code-based quality model for embedded systems: systematic development and validation by industrial projects, in: Proceedings of the 2012 IEEE 23rd International Symposium on Software Reliability Engineering (ISSRE), 2012, pp. 281–290.

[21] J. Friginal, D. de Andres, J.C. Ruiz, R. Moraes, Using dependability benchmarks to support ISO/IEC SQuaRE, in: Proceedings of the 2011 IEEE 17th Pacific Rim International Symposium on Dependable computing (PRDC), 2011, pp. 28–37.

[22] C.H. Loh, S. Lee, Predicting quality of object-oriented systems through a quality model based on design metrics and data mining techniques, in: Proceedings of the ICIME '09. International Conference on Information Management and Engineering, 2009, pp. 239–243.

[23] V. Srivastava, W. Farr, W. Ellis, Experience in the use of standard IEEE 982.1 on software programs., in: Proceedings of the International Conference and Workshop on Engineering of Computer-Based Systems 1997, 1997, p. 121-127..

[24] A. Jackson, HandBook-based high unit-value software reliability prediction method reliability and maintainability symposium, in: Proceedings of the Annual 2010, 2010, pp. 1–4.

[25] M. Sibisi, C.C. Van Waveren, A process framework for customising software quality models C, in: Proceedings of the AFRICON 2007, 2007, pp. 1–8.

[26] Q.Y. Dong, X.X. Fang, Z.A. Hong, Y.X. Bin, Analysis of contribution of conceptual model quality to software reliability, in: Proceedings of the International Conference on Computer Application and System Modeling (ICCASM, 2010), vol. 10, 2010, pp. 386–389.

[27] G. Low, V. Leenanuraksa, Software quality and case tools software technology and engineering practice, in: Proceedings of the STEP '99, 1999, pp. 142–150.

[28] H.J. Jung, H.S. Yang, Software reliability measurement use software reliability growth model in testing, in: Proceedings of the 2005 International Conference on Computational Science and Its Applications – Volume Part III Springer-Verlag, 2005, pp. 739–747.

[29] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and discussion, Requir. Eng. 11 (2006) 102–107.

[30] A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Dependable Secur. Comput. 1 (1) (2004) 11–33.

[31] IEEE- Std. 610.12 Standard Glossary of Software Engineering Terminology. 1990.

[32] J.D. Musa, Operational profiles in software-reliability engineering Software, IEEE 10 (2) (1993) 14–32.

[33] IEEE-Std 982.1. Standard Dictionary of Measures to Produce Reliable Software. 1988

[34] IEEE- Std 1633. IEEE Recommended Practice on Software Reliability. 2008

[35] T.L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, NY, 1980.

[36] T.L. Saaty, Fundamentals of Decision Making and Priority Theory with the Ahp Pittsburgh, RWS Publications, 2000.

[37] F. Alves Leite, et al., Introducing the analytic hierachy process: a mathematical homework, in: Proceedings of the 6th ISAHP, Berne, Switzerland, 2001, pp. 255–259.

[38] B. Beizer, Bug Taxonomy and Statistics, Appendix, Software Testing Techniques, Second ed., Van Nostrand Reinhold, New York, 1990.