ISSN 2007-9737

Predicting Software Product Quality: A Systematic Mapping Study

Sofia Ouhbi[1], Ali Idri[1], José Luis Fernández-Alemán[2], Ambrosio Toval[2]

[1]University Mohammed V, Software Project Management Research Team, ENSIAS, Rabat, Morocco

[2]University of Murcia, Department of Informatics and Systems, Faculty of Computer Science, Murcia, Spain

ouhbisofia@gmail.com, idri@ensias.ma, aleman@um.es, atoval@um.es

Abstract. Predicting software product quality (SPQ) is becoming a permanent concern during software life cycle phases. In this paper, a systematic mapping study was performed to summarize the existing SPQ prediction (SPQP) approaches in literature and to organize the selected studies according to seven classification criteria: SPQP approaches, research types, empirical types, data sets used in the empirical evaluation of these studies, artifacts, SQ models, and SQ characteristics. Publication channels and trends were also identified. After identifying 182 documents in ACM Digital Library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar, 69 papers were selected. The results show that the main publication source of the papers identified was conference. Data mining techniques are the most frequently SPQP approaches reported in literature. Solution proposal was the main research type identified. The majority of the papers selected were history-based evaluations using existing data which were mainly obtained from open source software projects and domain specific projects. Source code was the main artifact concerned with SPQP approaches. Well-known SQ models were hardly mentioned and reliability is the SQ characteristic through which SPQP was mainly achieved. SPQP-related subject seems to need more investigation from researchers and practitioners. Moreover, SQ models and standards need to be considered more in future SPQP research.

Keywords. Prediction, software product quality, systematic mapping study.

1 Introduction

Software quality (SQ) "is a complex concept. Because it means different things to different people, it is highly context-dependent" [56]. In the absence of a universally accepted definition of the SQ concept, the diversity of viewpoints makes the term ambiguous or difficult to understand. Software product quality (SPQ) could be defined as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" [1]. It is considered to be one of the most critical aspects as regards the success of software projects [3]. SPQ prediction (SPQP) plays a valuable role in software engineering practice. The objective of SPQP is to predict the SPQ level periodically and to indicate SPQ problems at an early stage [107]. Several measures, quality models, and standards with which to improve SQ have been proposed such as those of McCall [64], Boehm [15], Dromey [22], and the ISO/IEC 9126 standard [1] which has been replaced with ISO/IEC 25010 [4]. These models have some common SQ characteristics, i.e. efficiency, reliability, portability and maintainability [70].

This paper shows the results of a systematic mapping study which was performed to obtain an updated overview of the current approaches used in SPQP research. A systematic mapping study is a defined method with which to build a classification scheme and structure a field of interest [75]. Many systematic studies have been carried out in the SQ field, such as [25, 42, 68, 80, 94], but to the best of our knowledge, no systematic mapping study of SPQP approaches has been published to date.

Nine mapping questions are answered in this study and the papers which were selected after the search process are classified according to seven criteria. The main publication channels and trends were also identified. The results summarize the existing SPQP approaches and whether or not these approaches are based on well-known SQ models. Moreover, the results show if the SPQP is done through one or many SQ characteristics. The research types and empirical types that exist in literature are identified, and the data sets used in the evaluation of the approaches identified are listed. The results were analyzed, tabulated, and synthesized to provide both an updated and summarized view and a set of recommendations for researchers and practitioners.

The structure of this paper is as follows: Sect. 2 presents the research method used in this study. Sect. 3 reports the results obtained from the mapping study. Sect. 4 discusses the main findings, presents implications for researchers and practitioners, and outlines threats to validity. The conclusions and future work are presented in Sect. 5.

2 Research Methodology

The principal goal of a systematic mapping study is to provide an overview of a research area, and identify the quantity and type of research and results available within it. A mapping study differs from a systematic literature review (SLR) [20], in that the articles are not studied in detail.

A mapping process consists of three activities: the search for relevant publications, the definition of a classification scheme, and the mapping of publications [75].

2.1 Mapping Questions

This study aims to gain insight into the existing SPQP approaches. The systematic mapping study therefore addresses nine mapping questions (MQs). The nine MQs with the rationale motivating the importance of these questions are presented in Table 1. The search strategy and paper selection criteria were defined on the basis of them.

2.2 Search Strategy

The papers were identified by consulting the following sources [71, 72, 73]: IEEE Digital Library, ACM Digital Library, Springer and Science Direct. Google scholar was also used to seek literature in the field. IEEE and ACM are digital libraries recommended in software engineering because they cover an important number of journals and conferences [55]. The search string used to perform the automatic research in the digital libraries selected was formulated as follows:

"Software quality" **AND** (estimat* **OR** predict*) **AND** (technique* **OR** model* OR method* **OR** tool* **OR** framework* **OR** approach* **OR** process* **OR** learning **OR** data mining **OR** artificial intelligence **OR** pattern recognition **OR** analogy **OR** case based reasoning **OR** nearest neighbo* **OR** decision tree* **OR** classification tree* **OR** neural net* **OR** genetic programming **OR** genetic algorithm* **OR** bayesian belief network* **OR** bayesian net* **OR** association rule* **OR** support vector machine* **OR** regression **OR** fuzzy logic).

This search string was applied in the title, abstract, and keywords of the papers investigated to reduce the search results.

2.3 Paper Selection Criteria

Each paper was retrieved by one author. This author considered each paper title, abstract, and text, and then commented on whether the paper should be included, excluded, or if she was uncertain about it in the excel file. The evaluation of this selection was then made by another author in order to decide whether or not each paper should be included. Papers that were judged differently were discussed until an agreement was reached. The Kappa coefficient was 0.9 which, according to Landis and Koch [57], indicates an almost perfect agreement between the two assessments. The final selection was reviewed by the other two authors.

The first step after the application of the search string was to eliminate duplicate titles and titles clearly not related to the review. The inclusion criteria (IC) were limited to:

Table 1. Mapping questions

ID	Mapping question	Rationale
MQ1	Which publication channels are the main targets for SPQP research?	To identify where SPQP research can be found, in addition to the targets for the publication of future studies.
MQ2	How has the frequency of SPQP research dissemination changed over time?	To identify the publication trend of SPQP research over time.
MQ3	In which research types are SPQP papers classified?	To explore the different types of research in SPQP literature.
MQ4	Which approaches have been used for SPQP?	To identify the current approaches that have been used or proposed to predict SPQ.
MQ5	Are the SPQP selected studies empirically validated?	To discover the empirical types that have been used to validate SPQP approaches.
MQ6	What are the data sets that were used in SPQP literature?	To identify the data sets used in the evaluation of the empirical studies.
MQ7	Which artifacts have been reported in SPQP selected studies?	To identify the kind of artifacts that have been concerned with SPQP.
MQ8	What are the well-known SQ models that have been mentioned in SPQP literature?	To identify if the well-known SQ models have been used in the design of SPQP approaches.
MQ9	Which characteristics were used to predict SPQ?	To identify the SQ characteristics used to predict SPQ in literature.

IC The studies which address prediction or estimation of the quality of software product in overall or through SQ characteristics.

The studies that met at least one of the following exclusion criteria (EC) were excluded:

EC1 Papers that focus only on system quality not on SQ.

EC2 Papers whose subject was one or many SQ characteristics (e.g. maintainability, reliability) which were not used in the prediction of SQ.

EC3 Papers that were published before the nineties and after 2013.

Fig. 1 shows an overview of the search process and presents the number of studies remaining after each step of the selection process. In total, 119 papers were identified after the removal of duplicates. When the same paper appeared in more than one source, it was considered only once according to our search order. Thereafter, 50 studies were excluded based on the inclusion and exclusion criteria leaving for the final result 69 selected studies. The final results of the search in each library can be downloaded from the following website: https://goo.gl/gve9Hz.

2.4 Data Extraction Strategy

The selected studies were exploited to collect the data that would provide the set of possible answers to the MQs.

The publication source and channel of the papers respond to **MQ1**, while the publication year responds to **MQ2**.

MQ3. A research type can be classified as [20]:

— Evaluation research: existing SPQP approaches are implemented in practice and an evaluation of them is conducted.

— Solution proposal: an SPQP solution is proposed. This solution may be a new SPQP approach or a significant extension of an existing approach. The potential benefits and the applicability of the solution could be shown with an empirical study or a good argumentation.

— Other, e.g. experience paper, review.

MQ4. An approach can be classified as [75, 103]:

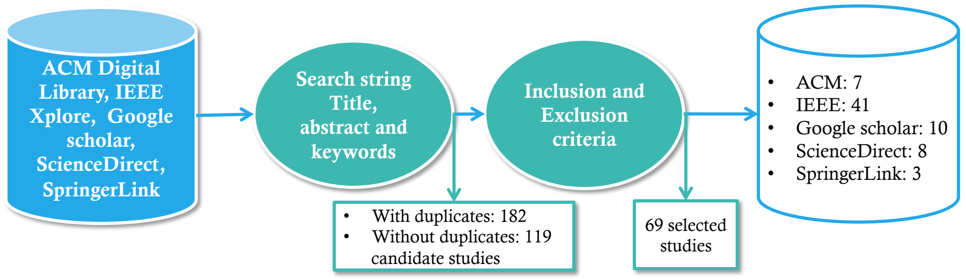
ACM Digital Library, IEEE Xplore, Google scholar, ScienceDirect, SpringerLink
Search string Title, abstract and keywords
Inclusion and Exclusion criteria
ACM: 7
IEEE: 41
Google scholar: 10
ScienceDirect: 8
SpringerLink: 3
With duplicates: 182
Without duplicates: 119 candidate studies
69 selected studies

Fig. 1. Selection process

— Data mining technique, such as [63]: regression analysis, clustering, bayesian network (BN), artificial neural network (ANN), fuzzy logic, learning technique, genetic algorithm (GA), genetic programming (GP), case-based reasoning (CBR), rough sets (RS), support vector machine (SVM), and decision tree (DT).

— Process: a series of actions, or functions leading to an SPQP result and performing operations on data.

— Method: a regular and systematic means of accomplishing SPQP.

— Tool: a means to accomplish SPQP tasks.

— Model: a system representation that allows SPQP to be investigated through a hierarchical structure.

— Framework: a real or conceptual structure intended to serve as a support or guide for SPQP.

— Other: e.g., guidelines.

MQ5. The selected studies can be classified as a [40]:

— Case study: an empirical inquiry that investigates an SPQP approach within its real-life context.

— Survey: a method for collecting quantitative information of an SPQP approach, e.g. a questionnaire.

— Experiment: an empirical method applied under controlled conditions to evaluate an SPQP approach.

— History-based evaluation: studies evaluating SPQP approaches in previously completed software projects.

— Theory: non-empirical research approaches or theoretical evaluation of an SPQP approach.

MQ6. The data can be retrieved from [103]:

— NASA [5] is a within-company database which include 13 publicly available data sets.

— ISBSG [37] is a non-free cross-company database which includes more than 5000 software projects gathered from different countries.

— PROMISE [65] is a repository of software engineering data, which is a collection of 20 publicly available datasets and tools.

— Open source software (OSS) project.

— Domain specific project (DSP): e.g., data from large development projects, from telecommunication software projects, or from a medical imaging system (MIS).

— Other: e.g., simulation data set.

MQ7. An artifact can be classified into [6, 84]: documentation, design module, source code, or other.

MQ8. A well-known SQ model can be [70]: McCall model [64], Boehm model [15], Dromey model [22], ISO/IEC 9126 standard [1], ISO/IEC 25010 standard [4], or other.

MQ9. An SQ characteristic can be classified into one of the internal and external quality characteristics originally proposed by ISO/IEC 9126 [1]: functionality, reliability, usability, efficiency, maintainability, portability, or other. This standard regroups common characteristics of the well-known SQ models and has influenced SQ research in the past decade [2, 6]. For these reasons, the ISO/IEC

Table 2. Publication channel

Publication channel	Total
Conference	37
Journal	23
Symposium	6
Workshop	3
Total	69

9126 [1] was chosen as a reference to answer this question rather than the ISO/IEC 25010 standard [4] which appeared in 2011 and does not cover the time frame of the studies selected.

2.5 Synthesis Method

The synthesis method was based on:

1. Counting the number of papers per publication channel and the number of papers found in each bibliographic source per year,
2. Counting the primary studies that are classified in each MQ's response,
3. Presenting charts and frequency tables for the classification results which have been used in the analysis,
4. Presenting in the discussion a narrative summary with which to recount the principal findings of this study.

3 Results

This section describes the results presented in Table 3.

3.1 MQ1. Publication Channels

Table 3 and Table 2 show the publication channels for SPQP research. Around 54% of the SPQP papers identified appeared in conferences, 33% were published in journals, 9% in symposia and only 4% in workshops. SPQP papers appeared in different journals and conferences. Note that the Software Quality Journal published only one paper on SPQP, despite the fact that it might be expected to attract more papers about SPQP approaches.

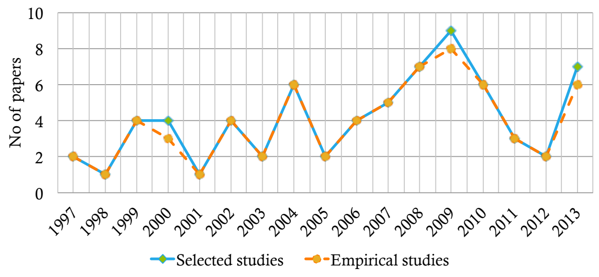

Fig. 2. Publication per year

3.2 MQ2. Publication Trend

Fig. 2 shows the publication trends of the papers selected. This figure also shows the trend of empirical studies selected in this paper. SPQP papers have been published since 1997. The interest in SPQP has increased in the last decade to reach a peak in 2009 after which it began to decrease to restart increasing in 2013. There is no obvious explanation for the decrease during the three years after 2009. The slight decrease observed for 2012 may be explained by the absence of a publication proposing an SPQP approach different than data mining techniques. Table 3 shows that non-data mining based SPQP approaches have been published every year since 2008, with the exception of the year 2012. Notice that the empirical trend is almost parallel to the SPQP trend in almost all the time slots between 1997 and 2013.

3.3 MQ3. Research Types

Fig. 3 shows the research type of the papers selected. The principal research type found is the solution proposal. Around 61% of the papers selected were solution proposal studies and except one review [8], the rest of papers were studies which were undertaken to evaluate SPQP approaches.

3.4 MQ4. SPQP Approaches

Fig. 3 also presents the SPQP approaches extracted from the papers selected. The approaches most frequently reported are those of data mining, principally regression [7, 21, 24, 31, 47, 50, 52, 67, 86, 91], fuzzy logic [12, 14, 36, 76, 82, 85, 98, 108],

Table 3. SQP result summary. Acronyms: Solution Proposal (SP), Evaluation Research (ER), Data Mining (DM), History-based Evaluation (HbE), Experiment (Ex), Case study (CS), Non-defined (ND), Accuracy (Ac), Reliability (Re), Stability (St), Efficiency (Ef), Maintainability (Ma), Fault-prone (FP).

Paper	P. Source	MQ1	MQ2	MQ3	MQ4	MQ5	MQ6	MQ7	MQ8	MQ9
[35]	COMPSACW	Workshop	2013	SP	Other	CS	OSS	Doc	ISO/IEC 9126	No
[81]	eKHOW	Conference	2013	SP	DM	HbE	DSP	ND	ISO/IEC 25010	All
[98]	IUP	Journal	2013	SP	DM	HbE	OSS	Code	No	No
[8]	ICISA	Conference	2013	Other	Model	Th	No	Code, Design, Doc	All	No
[82]	IJCSMC	Journal	2013	SP	DM	HbE	OSS	Code, Design	No	Re
[76]	Inform Sciences	Journal	2013	ER	DM	HbE	DSP	Code, Design, Doc	ISO/IEC 9126	Ma
[21]	KES	Conference	2013	ER	DM	CS	DSP	ND	Other	No
[13]	SAC	Symposium	2012	ER	DM	HbE	NASA	Code	No	Ac
[34]	ACCT	Conference	2012	ER	DM	CS	OSS	ND	No	FP
[11]	IST	Journal	2011	SP	Other	HbE	OSS	Code	No	St
[95]	WCRE	Conference	2011	SP	DM	HbE	PROMISE	Code, Design	No	FP
[14]	SIGSOFT Softw. Eng. Notes	Journal	2011	SP	DM	CS	OSS	Doc	No	Re
[17]	IST	Journal	2010	SP	DM	HbE	NASA	Code, Design	No	FP
[113]	ESWA	Journal	2010	ER	DM	HbE	NASA	Code, Design	No	Re
[92]	MIPRO	Conference	2010	SP	Other	CS	DSP	ND	ISO/IEC 9126, Other	FP
[44]	ICCCT	Conference	2010	ER	DM	HbE	NASA	Code	No	FP
[36]	MIPRO	Conference	2010	SP	DM	CS	DSP	Doc	ISO/IEC 9126	Re
[102]	WI-IAT	Conference	2010	ER	Model	HbE	ISBSG	ND	No	No
[10]	IST	Journal	2009	SP	DM	HbE	OSS	Code	No	St
[87]	IEEE T Syst Man Cy A	Journal	2009	ER	DM	HbE	NASA	Code	No	Re
[43]	ICMV	Conference	2009	SP	DM	HbE	NASA	Code	No	FP
[60]	ICIME	Conference	2009	SP	Model	HbE	Other	Code, Design	All	All
[28]	IRI	Conference	2009	ER	DM	CS	DSP	Code, Design	No	Re
[39]	PEITS	Conference	2009	SP	Process	Ex	OSS	Code	No	FP
[24]	MSR	Conference	2009	ER	DM	HbE	OSS	Code	No	St
[74]	ICRMS	Conference	2009	SP	Process	Th	No	ND	ISO/IEC 9126	Re
[30]	ARTCOM	Conference	2009	ER	DM	HbE	NASA	ND	No	Re
[69]	ESWA	Journal	2008	SP	Process	HbE	DSP	ND	No	No
[110]	IJCNN	Conference	2008	ER	DM	HbE	DSP	Code	No	Re
[83]	WoSQ	Workshop	2008	SP	DM	HbE	Other	Code, Design	No	Re
[16]	IS	Conference	2008	SP	DM	HbE	DSP	Code	No	Re
[111]	ICINIS	Conference	2008	SP	DM	HbE	NASA	Code, Design	No	Re
[91]	ICET	Conference	2008	ER	DM	HbE	PROMISE	Code	No	FP
[38]	PROMISE	Workshop	2008	ER	DM	HbE	NASA	Code, Design	No	FP
[101]	EASE	Conference	2007	SP	DM	Ex	DSP	Code	No	Re
[47]	IEEE T Reliab	Journal	2007	SP	DM	CS	DSP	Code	No	Re
[48]	IEEE T Reliab	Journal	2007	SP	DM	CS	DSP	ND	No	Re
[109]	ICNC	Conference	2007	SP	DM	HbE	Other	Design	ISO/IEC 9126	Re, Ef
[89]	SQJ	Journal	2007	ER	DM	HbE	NASA	Code	No	Re
[19]	GECCO	Conference	2006	SP	DM	HbE	OSS	Code	No	St
[66]	ICSEA	Conference	2006	SP	Process	HbE	NASA	ND	No	FP
[61]	MENSURA	Conference	2006	ER	Process	HbE	OSS	Code, Design	Other	FP
[7]	MENSURA	Conference	2006	ER	DM	HbE	OSS	Code	No	FP
[107]	ISSRE	Symposium	2005	SP	DM	HbE	DSP	Code	No	Re
[67]	ICSE	Conference	2005	SP	DM	HbE	OSS	Code, Doc	No	Re, Other
[51]	IEEE T Evolut Comput	Journal	2004	SP	DM	CS	DSP	Code	Other	Re
[114]	HASE	Symposium	2004	SP	DM	HbE	NASA	Code, Design	No	Re
[100]	ICTAI	Conference	2004	SP	DM	HbE	DSP	Code	No	FP
[90]	ICTAI	Conference	2004	ER	DM	HbE	NASA	Code, Design	No	FP
[54]	Adv Comput	Journal	2004	SP	DM	CS	DSP	Code	No	Re
[41]	SIGSOFT Softw. Eng. Notes	Journal	2004	ER	DM	Ex	OSS	Code	No	Re
[53]	ESE	Journal	2003	ER	DM	CS	DSP	Code	No	Re
[96]	ICSM	Conference	2003	ER	DM	HbE	OSS	Code, Design	No	Re
[77]	IJCNN	Conference	2002	SP	DM	HbE	OSS	Code	No	Ma, Ef
[18]	ICSM	Conference	2002	SP	DM	HbE	OSS	Code, Design	No	FP
[85]	COMPSAC	Conference	2002	SP	DM	HbE	OSS	Code	No	St
[52]	METRICS	Symposium	2002	ER	DM	CS	DSP	Code, Design	No	Re
[108]	The Comp. J.	Journal	2001	SP	DM	HbE	DSP	Code	No	No
[99]	IEEE Software	Journal	2000	SP	Method	Th	No	Code	No	Re
[27]	IJSEKE	Journal	2000	SP	DM	CS	DSP	Code	No	Re
[112]	ASSET	Symposium	2000	SP	DM	CS	DSP	ND	No	Re
[33]	APAQS	Conference	2000	SP	DM	HbE	DSP	Code	No	FP
[26]	IEEE T Software Eng	Journal	1999	SP	DM	HbE	NASA	Code, Design, Doc	No	Re
[86]	ISSRE	Symposium	1999	SP	DM	HbE	OSS	Code, Design	No	No
[50]	IJSEKE	Journal	1999	SP	DM	CS	DSP	Code	No	FP
[58]	CIE	Conference	1999	ER	DM	CS	DSP	Code	No	No
[49]	Computer	Journal	1998	ER	Process	CS	DSP	Code, Design, Doc	No	Re
[31]	ISSAT RQD	Conference	1997	ER	DM	HbE	DSP	Code	No	FP
[12]	SMC	Conference	1997	ER	DM	CS	DSP	Code, Design	No	Ac

and ANN [13, 41, 77, 96, 100, 109, 113] as shown in Fig. 4.

No tool has been proposed as the main solution to predict SPQ, however, supporting tools were developed to implement SPQP approaches such as in [14]. Processes, models, and a method were also identified. Some papers have proposed other approaches [11, 35, 92].

3.5 MQ5. Empirical Types

Fig. 3 also shows the empirical type used in the papers selected. Only 4% of the selected studies concerning SPQP approaches were not validated empirically. 64% of the papers selected had used existing data sets to evaluate SPQP approaches, 28% were validated through case studies while the

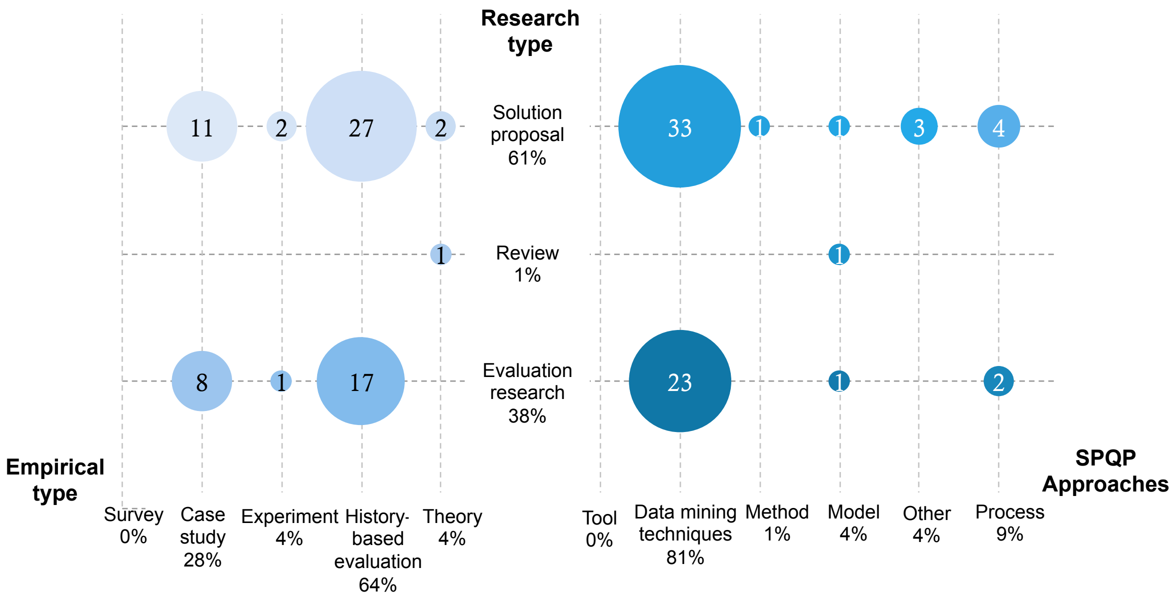

Fig. 3. Research types, SPQP approaches, and empirical types

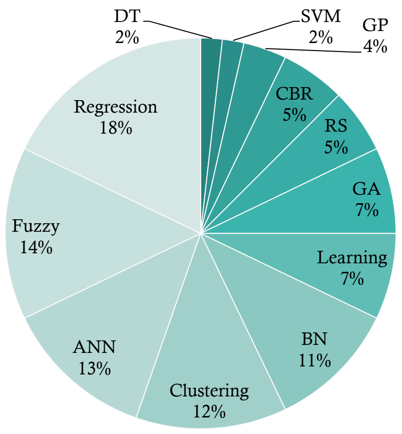

Fig. 4. Data mining techniques (56 studies)

rest were validated with experiments. None of the SPQP approaches were validated using surveys.

3.6 MQ6. Data Sets

Fig. 5 presents the data sets used to evaluate SPQP approaches. 96% of the studies selected have used data sets to predict SPQ. The data sets that were used were mainly retrieved from OSS projects, or DSPs, or NASA. 19% of data retrieved from DSPs are from MIS [16, 31, 33, 107, 110]. PROMISE was used in two studies [91, 95] and ISBSG data set was only used in one paper [102]. Few papers have used other data sets that were based on simulation data.

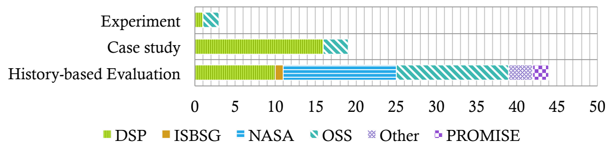

Fig. 5. Data sets (66 studies)

3.7 MQ7. Artifacts

Fig. 6 presents the software artifacts identified in the selected studies. This figure shows the number of papers which mention exclusively one artifact and the number of papers which mention more than one artifact. 84% papers mentioned a software artifact. Source code was discussed in 78% of the selected studies. 25% of the selected studies focussed exclusively on source code. Design module artifact was reported by 30% of SPQP research. Only 6% of the selected studies discussed documentation, mainly requirements documentations.

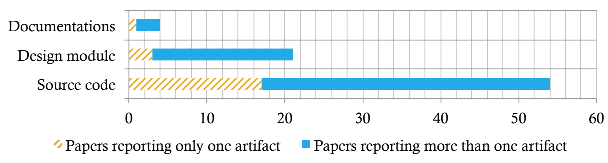

Fig. 6. Artifacts (58 studies)

3.8 MQ8. SQ Models

The results shown in Table 3 reveal that 17% of the selected SPQP papers cite a well-known SQ model. The principal model cited in these studies is the ISO/IEC 9126 standard [1]. McCall model [64], Dromey model [22], and Boehm model [15] are also cited. Note that only one paper [81] cited ISO/IEC 25010 [4]. Other papers mentioned other SQ models, such as FURPS/FURPS+ (i.e. functionality, usability, reliability, performance, and supportability) [60], module order modeling (MOM) [51], and the IEEE 12207 standard for software life cycle processes [92].

3.9 MQ9. SQ Characteristics

Table 3 shows that 13% of SPQP papers did not mention any SQ characteristic. This table shows also that the main characteristic used in the prediction of SPQ in the papers selected is reliability (including fault-proneness), followed by maintainability (including stability). The characteristic "satisfaction" was identified in the study [67], which is a characteristic for quality in use [1]. [77, 109] combined more than one SQ characteristic to achieve SQ, while [60, 81] have combined the totality of internal and external SQ characteristics. All the characteristics presented in Table 3, with the exception of fault-proneness, are actually part of the internal and external sub-characteristics of the ISO/IEC 9126 standard [1], such as stability which is a sub-characteristic of maintainability [1]. Fault-proneness directly affects reliability and was therefore considered in this study as a reliability sub-characteristic even though it was not included in the ISO/IEC 9126 standard [1].

4 Summary and Discussion

4.1 Principal Findings

SPQP topic is taken very seriously by researchers, as it has been observed in the amount of publications in recognized and stable journals and conferences. By the 1990s, major corporations recognized that billions of dollars were being spent and wasted each year on software that did not deliver the promised quality [79], thus leading researchers began to develop solutions and approaches with which to predict SQ. This explains the late interest in SPQP and the fact that the most frequent research type identified was that of solution proposal. The majority of SPQP approaches that were found in literature are data mining techniques, principally, regression and fuzzy logic, which is quite normal as there is an increasing interest in using data mining techniques in many software engineering subfields [106] and in other domains. The selected papers principally used existing data sets such as NASA [26, 59, 88] to evaluate SPQP solutions. Case studies were used to evaluate SPQP approaches and a few experiments were identified in the selected SPQP papers. This could be explained by the fact that more effort is needed to conduct experiments or case studies with new data and that it is easier to evaluate techniques using data from existing data sets repositories [32, 105]. Open source was the main source for data to validate SPQP approaches, and source code was the main artifact concerned by SPQP approaches, which indicates that researchers are more interested to predict internal SPQ. Only one study used the ISBSG repository, which may be due to the fact that this repository is not free and does not contain relevant SQ information.

Few researchers based their solutions on SQ models, particularly, the ISO/IEC 9126 standard [1], which may be explained by the fact that this standard is more recent than the other well-known SQ models. The ISO/IEC 25010 standard [4] is even more recent than the aforementioned standard, and appeared in 2011 to replace the ISO/IEC

9126 [1]. But of the 9 selected papers which were published after 2011, only one paper cited the ISO/IEC 25010 standard [4]. This lack of interest will change in the future, and authors may include this standard more frequently in their publications, while in the coming years the ISO/IEC 9126 [1] may gradually disappear from the SQ literature [93]. Note that the ISO/IEC 25010 quality model differs somewhat from the ISO/IEC 9126 quality model: security becomes a characteristic in ISO/IEC 25010 rather than a sub-characteristic for functionality as it was in ISO/IEC 9126, compatibility is added as a new characteristic in ISO/IEC 25010 and quality in use has five characteristics instead of the four characteristics of ISO/IEC 9126.

89% of SPQP studies that did not mention any well-known SQ models predicted SQ by using an SQ characteristic. This result could be justified by the fact that although well-known SQ models are sufficiently mature to provide a certain consensus as to what is desirable or not regarding SQ, they have several lacks for researchers as they are not very flexible, they are difficult to apply and require a lot of adaptation to particular situations [97] as they provide only the general SQ framework.

Reliability was the most frequently reported SQ characteristic in SPQP literature, and was in most cases achieved using fault-proneness. “Fault prevention is the initial defensive mechanism against unreliability. A fault which is never created costs nothing to fix” [62]. Since implementing faulty program modules may have a number of extremely negative consequences and is obviously undesirable, it is critical to ensure that all modules are error free [28, 87].

Complex mission-critical software systems depend heavily on the reliability of their software applications, and reliability is also a critical component in high-assurance software systems, such as those of telecommunications and medicine [114]. Some researchers predict SPQ through the sub-characteristics of the characteristics, which may be owing to the fact that these sub-characteristics are recommended for use as SQ metrics in the ISO/IEC 9126 standard. Few researchers have used characteristics other than reliability to predict SPQ, which could be explained by the difficulty involved in predicting them.

4.2 Implications of the Results

The findings of our systematic mapping study have implications for researchers and practitioners who work in the SQ domain, since this study will allow them to discover the existing SPQP approaches and techniques in the literature. Moreover, the presented empirical studies may provide an overview of the efficiency of each approach. More studies involving recent SQ models are needed to develop approaches that will meet SQ standards, particularly, the ISO/IEC 25010 standard [4]. Researchers should carry out more investigations into SQ reliability prediction, since the cost of software application failures is growing and the failures are increasingly impacting business performance [62]. The prediction of other characteristics, which have an impact on the emerging market of mobile software applications, should receive more attention from researchers. Researchers may also use other repositories for open research data sets in software engineering that the ones mentioned in this paper, such as SECOLD [46] which is the first online software ecosystem linked data platform of source code facts [45].

4.3 Threats to Validity

Construct validity: construct threats to validity in a mapping study are related to the identification of primary studies [9, 25]. In order to ensure that as many relevant primary studies as possible were included, different terms for SPQP approaches were added to the search string. However, the list might not have been complete, and additional or alternative terms such as “system quality” might have altered the final list of papers found [29]. Moreover, the references in the selected studies were not scanned to identify further studies. The final decision to select a study depended on the two authors who conducted the search process. If a disagreement arose between them, then a discussion took place until an agreement was reached.

Internal validity: internal validity deals with extraction and data analysis [9, 25]. Two authors carried out the data extraction and classification of the primary studies, while the other two authors reviewed the final results. The decision as to which data to collect and how to classify the papers therefore depended on the judgment of the two authors conducting the systematic mapping study. The Kappa coefficient was 0.9, reflecting a high level of agreement between the authors, which indicates a similar understanding of relevance, thus reducing this threat significantly.

Conclusion validity: in the case of a mapping study, this threat refers to such factors as missing studies and incorrect data extraction [9]. The aim is to control these factors so that a systematic mapping study can be performed by other researchers [23, 25, 29] who will draw the same conclusions [78]. In order to mitigate this threat, every step performed in the selection and data extraction activity was clearly described as discussed in the previous paragraphs. The traceability between the extracted data and the conclusions was strengthened through the direct generation of charts and frequency tables from the data by using a statistical package. In our opinion, slight differences based on the publication selection bias and misclassification would not alter the main conclusions drawn from the articles identified in our mapping study.

External validity: external validity is concerned with the generalization of this study [23, 104]. The systematic mapping results were considered with regard to the SPQP domain, and the validity of the conclusions drawn in this paper concerns only the SPQP context. This threat is not, therefore, present in this context. The results of this study may serve as a starting point for SQ researchers, and practitioners can search for and categorize additional papers accordingly.

5 Conclusions and Future Work

The overall goal of this study is to summarize the existing knowledge as regards SPQP approaches. Papers dealing with SPQP approaches from between 1990 and 2013 were identified. 69 papers were selected. The main publication sources of the identified papers were conferences and journals. The most interesting result of this study is that the approaches used to enhance SPQP that are most frequently reported in literature are those of data mining, and principally regression. Another interesting result is that few papers mention well-known SQ models in their research. More attention should be paid to the ISO/IEC 25010 standard [4] in the design of SPQP approaches. Reliability is the principal SQ characteristic through which the selected papers predict the overall SPQ, this characteristic was mainly achieved via fault-proneness. The results also demonstrated that the main concern of software researchers is to propose approaches with which to enhance SPQP, which was deduced from the fact that solution proposals were identified more frequently than evaluation research. Source code was the main artifact concerned by SPQP research. The majority of the selected papers were history-based evaluations using existing data sets. The data sets were mainly obtained from OSS projects and DSPs. Only a few papers extracted data from repositories.

This study could help practitioners to identify the approaches with which to enhance the SPQP in their projects, and it may also help researchers to identify both the data sets to be used in the evaluation of their studies and channels in which to publish their SPQP research results. Ongoing research is based on proposing an empirical method with which to evaluate SPQ.

Acknowledgments

This research is part of the project Software Project Management using Data Mining Techniques, (AP2010-2013), financed by Mohammed V University (Morocco), and part of the project GEODAS-REQ (TIN2012-37493-C03-02) financed by both the Spanish Ministry of Economy and Competitiveness and European FEDER funds. The mobility grant of Sofia Ouhbi is financed by the Mediterranean Office for Youth (MOY).

References

1. (**1999**). ISO/IEC Standard. ISO-9126 Software Product Evaluation - Quality Characteristics and Guidelines for Their Use.
2. (**2007**). Guide to the Software Quality Body of Knowledge (SQuBOK). Technical report, JUSE: The Union of Japanese Scientists and Engineers.
3. (**2009**). Standish-Group, CHAOS summary.
4. (**2011**). ISO/IEC 25010 standard. Systems and software engineering – Systems and software

Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

5. (**2013**). NASA Metrics Data Program (MDP) Repository.
6. **Abran, A. & Moore, J. W.** (**2004**). *Guide to the software engineering body of knowledge (SWEBOK).* IEEE Computer Society.
7. **Aggarwal, K., Singh, Y., Kaur, A., & Malhotra, R.** (**2006**). Improving logistic regression predictions of software quality using principal component analysis. *MENSURA*, pp. 226.
8. **Al-Jamimi, H. A. & Ahmed, M.** (**2013**). Machine learning-based software quality prediction models: state of the art. *International Conference on Information Science and Applications (ICISA)*, IEEE, pp. 1–4.
9. **Ampatzoglou, A., Charalampidou, S., & Stamelos, I.** (**2013**). Research state of the art on GoF design patterns: A mapping study. *Journal of Systems and Software*.
10. **Azar, D., Harmanani, H., & Korkmaz, R.** (**2009**). A hybrid heuristic approach to optimize rule-based software quality estimation models. *Information and Software Technology*, Vol. 51, No. 9, pp. 1365–1376.
11. **Azar, D. & Vybihal, J.** (**2011**). An ant colony optimization algorithm to improve software quality prediction models: Case of class stability. *Information and Software Technology*, Vol. 53, No. 4, pp. 388–393.
12. **Baisch, E. & Liedtke, T.** (**1997**). Comparison of conventional approaches and soft-computing approaches for software quality prediction. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1045–1049.
13. **Banthia, D. & Gupta, A.** (**2012**). Investigating fault prediction capabilities of five prediction models for software quality. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, ACM, New York, NY, USA, pp. 1259–1261.
14. **Bhatia, N. & Kapoor, N.** (**2011**). Fuzzy cognitive map based approach for software quality risk analysis. *SIGSOFT Software Engineering Notes*, Vol. 36, No. 6, pp. 1–9.
15. **Boehm, B. W., Brown, J. R., Kaspar, H., & Lipow, M.** (**1978**). *Characteristics of software quality*. TRW Softw. Technol. North-Holland, Amsterdam.
16. **Bouguila, N., Wang, J. H., & Ben Hamza, A.** (**2008**). A bayesian approach for software quality prediction. *4th International IEEE Conference Intelligent Systems*, volume 2, pp. 49–54.
17. **Bouktif, S., Ahmed, F., Khalil, I., & Antoniol, G.** (**2010**). A novel composite model approach to improve software quality prediction. *Information and Software Technology*, Vol. 52, No. 12, pp. 1298–1311.
18. **Bouktif, S., Kegl, B., & Sahraoui, H.** (**2002**). Combining software quality predictive models: An evolutionary approach. *International Conference on Software Maintenance*, pp. 385–392.
19. **Bouktif, S., Sahraoui, H., & Antoniol, G.** (**2006**). Simulated annealing for improving software quality prediction. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, ACM, New York, NY, USA, pp. 1893–1900.
20. **Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M.** (**2007**). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, Vol. 80, No. 4, pp. 571–583.
21. **do Prado, H. A., Bianchi Campos, F., Ferneda, E., Nunes Cornelio, N., & Haendchen Filho, A.** (**2013**). Prediction of software quality based on variables from the development process. *Proceedings of the 16th International Conference on Knowledge Engineering, Machine Learning and Lattice Computing with Applications*, KES'12, Springer-Verlag, Berlin, Heidelberg, pp. 71–77.
22. **Dromey, R. G.** (**1996**). Cornering the Chimera. *IEEE Software*, Vol. 13, No. 1, pp. 33–43.
23. **Easterbrook, S., Singer, J., Storey, M.-A., & Damian, D.** (**2008**). Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer, pp. 285–311.
24. **Ekanayake, J., Tappolet, J., Gall, H. C., & Bernstein, A.** (**2009**). Tracking concept drift of software projects using defect prediction quality. *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories*, MSR '09, IEEE Computer Society, Washington, DC, USA, pp. 51–60.
25. **Elberzhager, F., Münch, J., & Nha, V. T. N.** (**2012**). A systematic mapping study on the combination of static and dynamic quality assurance techniques. *Information and Software Technology*, Vol. 54, No. 1, pp. 1–15.

26. **Fenton, N. E. & Neil, M.** (**1999**). A critique of software defect prediction models. *IEEE Transactions Software Engineering*, Vol. 25, No. 5, pp. 675–689.

27. **Ganesan, K., M, K. T., & B, A. E.** (**2000**). Case-based software quality prediction. *International Journal of Software Engineering and Knowledge Engineering*, Vol. 10, pp. 139–152.

28. **Gao, K., Khoshgoftaar, T. M., & Wang, H.** (**2009**). An empirical investigation of filter attribute selection techniques for software quality classification. *Proceedings of the 10th IEEE international conference on Information Reuse & Integration*, IRI'09, IEEE Press, Piscataway, NJ, USA, pp. 272–277.

29. **Garousi, V., Mesbah, A., Betin-Can, A., & Mirshokraie, S.** (**2013**). A systematic mapping study of web application testing. *Information and Software Technology*, Vol. 55, No. 8, pp. 1374–1396.

30. **Gayatri, N., Nickolas, S., Reddy, A. V., & Chitra, R.** (**2009**). Performance analysis of datamining algorithms for software quality prediction. *Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing*, ARTCOM '09, IEEE Computer Society, Washington, DC, USA, pp. 393–395.

31. **Gokhale, S. S. & Lyu, M. R.** (**1997**). Regression tree modeling for the prediction of software quality. *International Conference on Reliability and Quality in Design*, pp. 31–36.

32. **González-Barahona, J. M. & Robles, G.** (**2012**). On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering*, Vol. 17, No. 1-2, pp. 75–89.

33. **Guo, P. & Lyu, M. R.** (**2000**). Software quality prediction using mixture models with EM algorithms. *Proceedings of the The First Asia-Pacific Conference on Quality Software (APAQS'00)*, APAQS '00, IEEE Computer Society, Washington, DC, USA, pp. 69–78.

34. **Gupta, D., Goyal, V. K., & Mittal, H.** (**2012**). Analysis of clustering techniques for software quality prediction. *International Conference on Advanced Computing and Communication Technologies*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 6–9.

35. **Hmood, A. & Rilling, J.** (**2013**). Analyzing and predicting software quality trends using financial patterns. *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, IEEE, pp. 481–486.

36. **Hribar, L. & Duka, D.** (**2010**). Software component quality prediction using KNN and Fuzzy logic. *Proceedings 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics*, volume 2, pp. 153–159.

37. **ISBSG** (**2013**). International Software Benchmarking Standards Group.

38. **Jiang, Y., Cuki, B., Menzies, T., & Bartlow, N.** (**2008**). Comparing design and code metrics for software quality prediction. *Proceedings of the 4th international workshop on Predictor models in software engineering*, PROMISE '08, ACM, New York, NY, USA, pp. 11–18.

39. **Jin, C., Jin, S.-W., Ye, J.-M., & Zhang, Q.-G.** (**2009**). Quality prediction model of object-oriented software system using computational intelligence. *International Conference on Power Electronics and Intelligent Transportation System*, pp. 120–123.

40. **Jorgensen, M. & Shepperd, M.** (**2007**). A systematic review of software development cost estimation studies. *IEEE Transactions Software Engineering*, Vol. 33, No. 1, pp. 33–53.

41. **Kanmani, S., Uthariaraj, V. R., Sankaranarayanan, V., & Thambidurai, P.** (**2004**). Object oriented software quality prediction using general regression neural networks. *SIGSOFT Software Engineering Notes*, Vol. 29, No. 5, pp. 1–6.

42. **Karg, L. M., Grottke, M., & Beckhaus, A.** (**2011**). A systematic literature review of software quality cost research. *Journal of Systems and Software*, Vol. 84, No. 3, pp. 415–427.

43. **Kaur, A., Sandhu, P. S., & Bra, A. S.** (**2009**). Early software fault prediction using real time defect data. *Proceedings of the 2009 Second International Conference on Machine Vision*, ICMV '09, IEEE Computer Society, Washington, DC, USA, pp. 242–245.

44. **Kaur, D., Kaur, A., Gulati, S., & Aggarwal, M.** (**2010**). A clustering algorithm for software fault prediction. *International Conference on Computer and Communication Technology*, pp. 603–607.

45. **Keivanloo, I., Forbes, C., Hmood, A., Erfani, M., Neal, C., Peristerakis, G., & Rilling, J.** (**2012**). A linked data platform for mining software repositories. *9th IEEE Working Conference on Mining Software Repositories (MSR)*, IEEE, pp. 32–35.

46. **Keivanloo, I., Forbes, C., Rilling, J., & Charland, P.** (**2011**). Towards sharing source code facts using linked data. *Proceedings of the 3rd International*

Workshop on Search-Driven Development: Users, Infrastructure, Tools, and Evaluation, SUITE '11, ACM, New York, NY, USA, pp. 25–28.

47. **Khoshgoftaar, T. & Gao, K.** (**2007**). Count models for software quality estimation. *IEEE Transactions on Reliability*, Vol. 56, pp. 212–222.

48. **Khoshgoftaar, T. & Liu, Y.** (**2007**). A multi-objective software quality classification model using genetic programming. *IEEE Transactions on Reliability*, Vol. 56, No. 2, pp. 237–245.

49. **Khoshgoftaar, T. M., Allen, E. B., Halstead, R., Trio, G. P., & Flass, R. M.** (**1998**). Using process history to predict software quality. *Computer*, Vol. 31, No. 4, pp. 66–72.

50. **Khoshgoftaar, T. M., Allen, E. B., Jones, W. D., & Hudepohl, J. P.** (**1999**). Data mining for predictors of software quality. *International Journal of Software Engineering and Knowledge Engineering*, Vol. 09, No. 05, pp. 547–563.

51. **Khoshgoftaar, T. M., Liu, Y., & Seliya, N.** (**2004**). A multiobjective module-order model for software quality enhancement. *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 6, pp. 593–608.

52. **Khoshgoftaar, T. M. & Seliya, N.** (**2002**). Tree-based software quality estimation models for fault prediction. *Proceedings of the 8th International Symposium on Software Metrics*, METRICS '02, IEEE Computer Society, Washington, DC, USA, pp. 203–214.

53. **Khoshgoftaar, T. M. & Seliya, N.** (**2003**). Fault prediction modeling for software quality estimation: Comparing commonly used techniques. *Empirical Software Engineering*, Vol. 8, No. 3, pp. 255–283.

54. **Khoshgoftaar, T. M. & Seliya, N.** (**2004**). Software quality estimation with case-based reasoning. volume 62 of *Advances in Computers*. Elsevier, pp. 249–291.

55. **Kitchenham, B. & Brereton, P.** (**2013**). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, Vol. 55, No. 12, pp. 2049 – 2075.

56. **Kitchenham, B. & Pfleeger, S. L.** (**1996**). Software quality: The elusive target. *IEEE Software*, Vol. 13, No. 1, pp. 12–21.

57. **Landis, J. & Koch, G.** (**1977**). The measurement of observer agreement for categorical data. *Biometrics*, Vol. 33, pp. 159–174.

58. **Lewis Nigel, D. C.** (**1999**). Assessing the evidence from the use of spc in monitoring, predicting & improving software quality. *Comput. Ind. Eng.*, Vol. 37, No. 1-2, pp. 157–160.

59. **Liu, Y., Khoshgoftaar, T. M., & Seliya, N.** (**2010**). Evolutionary optimization of software quality modeling with multiple repositories. *IEEE Transactions Software Engineering*, Vol. 36, No. 6, pp. 852–864.

60. **Loh, C. H. & Lee, S. P.** (**2009**). Predicting quality of object-oriented systems through a quality model based on design metrics and data mining techniques. *Proceedings of the 2009 International Conference on Information Management and Engineering*, ICIME '09, IEEE Computer Society, Washington, DC, USA, pp. 239–243.

61. **Lounis, H., Abdi, M., & Yazid, H.** (**2006**). Predicting quality attributes via machine-learning algorithms. *MENSURA*, pp. 72.

62. **Lyu, M. R.** (**2007**). Software reliability engineering: A roadmap. *2007 Future of Software Engineering*, FOSE '07, IEEE Computer Society, Washington, DC, USA, pp. 153–170.

63. **Margaret, H. D.** (**2003**). Data mining introductory and advanced topics. *Pearsons Education Inc*.

64. **McCall, J. A.** (**2002**). *Quality Factors*. John Wiley & Sons, Inc.

65. **Menzies, T., Caglayan, B., Kocaguneli, E., Krall, J., Peters, F., & Turhan, B.** (**2012**). The PROMISE Repository of empirical software engineering data.

66. **Mertik, M., Lenic, M., Stiglic, G., & Kokol, P.** (**2006**). Estimating software quality with advanced data mining techniques. *Proceedings of the International Conference on Software Engineering Advances*, ICSEA '06, IEEE Computer Society, Washington, DC, USA, pp. 19.

67. **Mockus, A., Zhang, P., & Li, P. L.** (**2005**). Predictors of customer perceived software quality. *Proceedings of the 27th international conference on Software engineering*, ICSE '05, ACM, New York, NY, USA, pp. 225–233.

68. **Montagud, S., Abrahão, S., & Insfran, E.** (**2012**). A systematic review of quality attributes and measures for software product lines. *Software Quality Journal*, Vol. 20, No. 3-4, pp. 425–486.

69. **Moreno García, M. N., Román, I. R., García Peñalvo, F. J., & Bonilla, M. T.** (**2008**). An association rule mining method for estimating the impact of project management policies on software quality, development time and effort. *Expert Syst. Appl.*, Vol. 34, No. 1, pp. 522–529.

ISSN 2007-9737

70. Ortega, M., Pérez, M., & Rojas, T. (**2003**). Construction of a systemic quality model for evaluating a software product. *Software Quality Control*, Vol. 11, No. 3, pp. 219–242.

71. Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (**2013**). Software quality requirements: a systematic mapping study. *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, volume 1, IEEE, pp. 231–238.

72. Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (**2014**). Evaluating software product quality: A systematic mapping study. *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, IEEE, pp. 141–151.

73. Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (**2015**). Requirements engineering education: a systematic mapping study. *Requirements Engineering*, Vol. 20, No. 2, pp. 119–138.

74. Peng, W., Yao, L., & Miao, Q. (**2009**). An approach of software quality prediction based on relationship analysis and prediction model. *International Conference on Reliability, Maintainability and Safety*, pp. 713–717.

75. Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (**2008**). Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*, Blekinge Institute of Technology, Bari, Italy, pp. 71–80.

76. Pizzi, N. J. (**2013**). A fuzzy classifier approach to estimating software quality. *Information Sciences*.

77. Pizzi, N. J., Summers, A. R., & Pedrycz, W. (**2002**). Software quality prediction using median-adjusted class labels. *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pp. 2405–2409.

78. Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M., & Beecham, S. (**2012**). Tools used in global software engineering: A systematic mapping review. *Information and Software Technology*, Vol. 54, No. 7, pp. 663–685.

79. Pressman, R. (**2009**). Software engineering: A practitioner's approach.

80. Radjenović, D., Heričko, M., Torkar, R., & Živkovič, A. (**2013**). Software fault prediction metrics: A systematic literature review. *Information and Software Technology*, Vol. 55, No. 8, pp. 1397–1418.

81. Radlinski, L. (**2013**). An expert-driven bayesian network model for simulating and predicting software quality. *eKNOW 2013, The Fifth International Conference on Information, Process, and Knowledge Management*, pp. 26–31.

82. Rana, S. & Yadav, R. K. (**2013**). A fuzzy improved association mining approach to estimate software quality. *International Journal of Computer Science and Mobile Computing*.

83. Rana, Z. A., Shamail, S., & Awais, M. M. (**2008**). Towards a generic model for software quality prediction. *Proceedings of the 6th international workshop on Software quality*, WoSQ '08, ACM, New York, NY, USA, pp. 35–40.

84. Rastkar, S., Murphy, G. C., & Murray, G. (**2010**). Summarizing software artifacts: a case study of bug reports. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, volume 1 of *ICSE '10*, ACM, New York, NY, USA, pp. 505–514.

85. Sahraoui, H. A., Boukadoum, M. A., Chawiche, H. M., Mai, G., & Serhani, M. (**2002**). A fuzzy logic framework to improve the performance and interpretation of rule-based quality prediction models for oo software. *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment*, COMPSAC '02, IEEE Computer Society, Washington, DC, USA, pp. 131–138.

86. Schneidewind, N. F. & Nikora, A. P. (**1999**). Predicting deviations in software quality by using relative critical value deviation metrics. *Proceedings of the 10th International Symposium on Software Reliability Engineering*, ISSRE '99, IEEE Computer Society, Washington, DC, USA, pp. 136–146.

87. Seiffert, C., Khoshgoftaar, T. M., & Van Hulse, J. (**2009**). Improving software-quality predictions with data sampling and boosting. *Trans. Sys. Man Cyber. Part A*, Vol. 39, No. 6, pp. 1283–1294.

88. Seliya, N. & Khoshgoftaar, T. M. (**2007**). Software quality analysis of unlabeled program modules with semisupervised clustering. *Trans. Sys. Man Cyber. Part A*, Vol. 37, No. 2, pp. 201–211.

89. Seliya, N. & Khoshgoftaar, T. M. (**2007**). Software quality estimation with limited fault data: a semi-supervised learning perspective. *Software Quality Control*, Vol. 15, No. 3, pp. 327–344.

90. Seliya, N., Khoshgoftaar, T. M., & Zhong, S. (**2004**). Semi-supervised learning for software quality estimation. *Proceedings of the 16th IEEE*

International Conference on Tools with Artificial Intelligence, ICTAI '04, IEEE Computer Society, Washington, DC, USA, pp. 183–190.

91. **Shafi, S., Hassan, S. M., Arshaq, A., Khan, M. J., & Shamail, S.** (**2008**). Software quality prediction techniques: A comparative analysis. *4th International Conference on Emerging Technologies*, pp. 242–246.

92. **Sinovcic, I. & Hribar, L.** (**2010**). How to improve software development process using mathematical models for quality prediction and elements of six sigma methodology. *MIPRO*, pp. 388–395.

93. **SOLEY, B., Richard Mark et CURTIS** (**2013**). The Consortium for IT Software Quality (CISQ). *Software Quality. Increasing Value in Software and Systems Development*, pp. 3–9.

94. **Tahir, A. & MacDonell, S. G.** (**2012**). A systematic mapping study on dynamic metrics and software quality. *28th IEEE International Conference on Software Maintenance (ICSM)*, IEEE, pp. 326–335.

95. **Tan, X., Peng, X., Pan, S., & Zhao, W.** (**2011**). Assessing software quality by program clustering and defect prediction. *Proceedings of the 2011 18th Working Conference on Reverse Engineering*, WCRE '11, IEEE Computer Society, Washington, DC, USA, pp. 244–248.

96. **Thwin, M. M. T. & Quah, T.-S.** (**2005**). Application of neural networks for software quality prediction using object-oriented metrics. *Journal of Systems and Software*, Vol. 76, No. 2, pp. 147–156.

97. **Vanderose, B. & Habra, N.** (**2011**). Tool-support for a model-centric quality assessment: QuaTALOG. *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 263–268.

98. **Verma, G. & Tomar, P.** (**2013**). Predicting quality using fuzzy model on object-oriented software design. *IUP Journal of Computer Sciences*, Vol. 7, No. 4.

99. **Voas, J.** (**2000**). Can chaotic methods improve software quality predictions? *IEEE Software*, Vol. 17, No. 5, pp. 20–22.

100. **Wang, Q., Yu, B., & Zhu, J.** (**2004**). Extract rules from software quality prediction model based on neural network. *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '04, IEEE Computer Society, Washington, DC, USA, pp. 191–195.

101. **Wang, Q., Zhu, J., & Yu, B.** (**2007**). Feature selection and clustering in software quality prediction. *Proceedings of the 11th international conference on Evaluation and Assessment in Software Engineering*, EASE'07, British Computer Society, Swinton, UK, UK, pp. 21–32.

102. **Wang, X., Zhang, Y., Zhang, L., & Shi, Y.** (**2010**). A knowledge discovery case study of software quality prediction: Isbsg database. *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 3 of *WI-IAT '10*, IEEE Computer Society, Washington, DC, USA, pp. 219–222.

103. **Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C.** (**2012**). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, Vol. 54, No. 1, pp. 41–59.

104. **Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A.** (**2000**). *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA.

105. **Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A.** (**2012**). *Experimentation in software engineering*. Springer.

106. **Xie, T., Thummalapenta, S., Lo, D., & Liu, C.** (**2009**). Data mining for software engineering. *Computer*, Vol. 42, No. 8, pp. 55–62.

107. **Xing, F., Guo, P., & Lyu, M. R.** (**2005**). A novel method for early software quality prediction based on support vector machine. *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering*, ISSRE '05, IEEE Computer Society, Washington, DC, USA, pp. 213–222.

108. **Xu, Z. & Khoshgoftaar, T. M.** (**2001**). Software quality prediction for high-assurance network telecommunications systems. *The Computer Journal*, Vol. 44, No. 6, pp. 557–568.

109. **Yang, B., Yao, L., & Huang, H.-Z.** (**2007**). Early software quality prediction based on a fuzzy neural network model. *Proceedings of the Third International Conference on Natural Computation*, volume 1 of *ICNC '07*, IEEE Computer Society, Washington, DC, USA, pp. 760–764.

110. **Yang, B., Yin, Q., Xu, S., & Guo, P.** (**2008**). Software quality prediction using affinity propagation algorithm. *IEEE International Joint Conference on Neural Networks*, pp. 1891–1896.

111. Yang, W. & Li, L. (**2008**). A new method to predict software defect based on rough sets. *Proceedings of the 2008 First International Conference on Intelligent Networks and Intelligent Systems*, ICINIS '08, IEEE Computer Society, Washington, DC, USA, pp. 135–138.

112. Yuan, X., Khoshgoftaar, T. M., Allen, E. B., & Ganesan, K. (**2000**). An application of fuzzy clustering to software quality prediction. *Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00)*, ASSET '00, IEEE Computer Society, Washington, DC, USA, pp. 85–90.

113. Zheng, J. (**2010**). Cost-sensitive boosting neural networks for software defect prediction. *Expert Syst. Appl.*, Vol. 37, No. 6, pp. 4537–4543.

114. Zhong, S., Khoshgoftaar, T. M., & Seliya, N. (**2004**). Unsupervised learning for expert-based software quality estimation. *Proceedings of the Eighth IEEE international conference on High assurance systems engineering*, HASE'04, IEEE Computer Society, Washington, DC, USA, pp. 149–155.

Sofia Ouhbi is a Ph.D. scholar at ENSIAS, University Mohammed V, Rabat, Morocco. She is also a Ph.D. student at the University of Murcia, Spain. She received an engineering degree in IT from INPT, Rabat, Morocco, in 2009. She also received a Master degree from the University of Lorraine, France, in 2013. She had two year experience working at Hewlett-Packard as Service Partner Manager for Southern and Eastern Africa. Her research interests include software quality, requirements engineering, and e-health. She has published more than 10 papers in peer reviewed conferences and journals.

Ali Idri is a Professor at the Computer Science and Systems Analysis School (ENSIAS, University Mohammed V, Rabat, Morocco). He received DEA (Master)(1994) and Doctorate of 3rd Cycle (1997) degrees in Computer Science, both from the University Mohamed V of Rabat. He received his Ph.D. (2003) in Cognitive Computer Sciences from the University of Quebec at Montreal. He has been the head of the Software Project Management research team since 2010. He serves as a member of the program committee of major international journals and conferences. His research interests include software effort/cost estimation, software metrics, software quality, computational intelligence in software engineering, data mining, e-health. He has published more than 90 papers in several international journals and conferences.

José Luis Fernández Alemán is an Associate Professor at the University of Murcia (Spain), where he is a member of the Software Engineering Research Group. He received his B.Sc. (Hons.) in 1994 and his Ph.D. in 2002, both in Computer Science from the University of Murcia. He has published more than 20 JCR papers in the areas of software engineering and requirements engineering, and their application to the fields of e-health and e-learning. His publications include articles in such highly ranked international journals as Journal of Medical Internet Research, Journal of Biomedical Informatics, IEEE Computer, IEEE Software, and IEEE Transactions on Software Engineering. Currently, his main research interest is m-health and m-learning, and their application to computer science, medicine, and nursing. He has contributed to many Spanish-funded research projects and technology transfer contracts whose topics were related to software engineering.

Ambrosio Toval Álvarez is a Full Professor at the University of Murcia in Spain. He received his B.Sc. in Mathematics from the University Complutense of Madrid and his Ph.D. in Computer Science (cum laude) from the Technical University of Valencia (both in Spain). He is involved in a variety of applied research and development projects with industry and conducts research and technology transfer in the areas of requirements engineering processes and tools, privacy and security requirements, and applications in the e-health, e-learning, and mobile development domains. He has published in the same topics in such international journals as IEEE Software, IST, REJ, Computer Standards & Interfaces, IET, IJIS, etc. Dr. Toval is currently the Head of the Software Engineering Research Group at the University of Murcia.

Article received on 14/04/2014; accepted on 10/06/2015.
Corresponding author is Sofia Ouhbi.