2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement

Evaluating Software Product Quality: A Systematic Mapping Study

Sofia Ouhbi, Ali Idri
Software Project Management research team
ENSIAS, University Mohammed V Souissi
Rabat, Morocco
ouhbisofia@gmail.com, idri@ensias.ma

Jose Luis Fernández-Alemán, Ambrosio Toval
Dept. Informatica y Sistemas
University of Murcia
Murcia, Spain
aleman@um.es, atoval@um.es

***Abstract*— Evaluating software product quality (SPQ) is an important task to ensure the quality of software products. In this paper a systematic mapping study was performed to summarize the existing SPQ evaluation (SPQE) approaches in literature and to classify the selected studies according to seven classification criteria: SPQE 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. 57 papers were selected. The results show that the main publication sources of the papers identified were journals. Data mining techniques are the most frequently approaches reported in literature. Solution proposals were the main research type identified. The majority of the selected papers 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 artifacts used by SPQE approaches. Well-known SQ models were mentioned by half of the selected papers and reliability is the SQ characteristic through which SPQE was mainly achieved. SPQE-related subjects seem to attract more interest from researchers since the past years.**

I. Introduction

According to ISO/IEC 9126 standard [2] software product quality (SPQ) could be defined as "the totality of characteristics of an entity that bear on it's ability to satisfy stated and implied needs". SPQ evaluation (SPQE) play a valuable role in software engineering practice. The objective SPQE is to help to check whether the expected software quality (SQ) has been achieved [7]. Several SQ models and standards with which to improve SPQ have been proposed such as those of McCall [4], Boehm [5], Dromey [6] and the ISO/IEC 9126 standard [2] which has been replaced with ISO/IEC 25010 [7]. These models have some common SQ characteristics, such as: efficiency, reliability, portability and maintainability [8]. In the literature [9]–[11] various approaches have been applied for SPQE, such as data mining techniques and models.

This paper presents the results of a systematic mapping study that was performed to obtain the current SPQE approaches used in SPQE literature. A systematic mapping study is a defined method with which to build a classification scheme and structure a field of interest [13]. A mapping study differs from a systematic literature review [12], in that the articles are not studied in detail. Many systematic studies have been carried out in the SQ field, such as [14], [15], but to the best of our knowledge, no systematic mapping study of SPQE approaches has been published to date. Nine mapping questions are answered in this study and the papers that 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 SPQE approaches and whether or not these approaches are based on well-known SQ models. Moreover, the results show if the SPQE 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. II presents the research method used in this study. Sect. III reports the results obtained from the systematic mapping study. Sect. IV discusses the main finding, presents implications for researchers and practitioners and outlines threats to validity. The conclusions and future work are presented in Sect. V.

II. Research methodology

The systematic mapping study principal goal is to provide an overview of a research area, and identify the quantity and type of research and results available within it. A mapping process consists of three activities: the search for relevant publications, the definition of a classification scheme and the mapping of publications [13]. This method focuses on classification, conducting a thematic analysis and identifying publication fora.

A. Mapping questions

This study aims to gain insight into the existing SPQE 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 I. The search strategy and paper selection criteria were defined on the basis of them.

B. Search strategy

The papers were identified by consulting the following sources: IEEE Digital Library, ACM Digital Library, Science Direct and SpringLink. Google scholar was also used to seek literature in the field. The search was done in January 2014. The search string used to perform the automatic research in the digital libraries selected was formulated as follows:

978-1-4799-4174-2/14 $31.00 © 2014 IEEE
DOI 10.1109/IWSM.Mensura.2014.30

TABLE I: Mapping questions

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

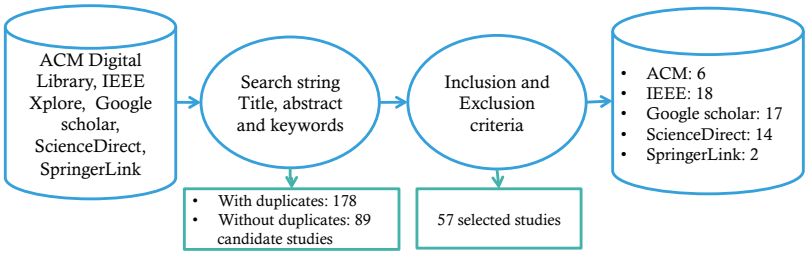

Fig. 1: Selection process

"Software quality" **AND** (evaluat* **OR** measur* **OR** assess*) **AND** (technique* **OR** approach* **OR** method* **OR** model* **OR** framework* **OR** tool* **OR** data mining **OR** learning **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.

C. Paper selection criteria

Each paper was retrieved by one author and the information about it was filed in the excel file shown in Table II.

TABLE II: Fields on the selection sheet

Title
Authors
Source
Year
Abstract
Keywords
Comments
To be included
To be excluded
Uncertain

The author responsible for retrieving the papers 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, measured before the disucussion, was 0.9 which, according to Landis and Koch [16], 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:

IC The studies that address evaluation, measurement or assessment 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 on system quality not on SPQ.
EC2 Papers whose subject was one or many SQ characteristics (e.g. maintainability, reliability) which were not used in the evaluation of SPQ.
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, 89 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, 32 studies were excluded based on the inclusion and exclusion criteria leaving for the final result 57 selected studies.

D. Data extraction strategy

The selected studies were exploited to collect the data that would provide the set of possible answers to the MQs. Table III shows the data extraction form, which was created as an excel sheet and filled in by one author for each of the papers selected.

TABLE III: Fields on the data extraction form

Paper ID
Title
Classification
(MQ1) Publication channel
(MQ2) Publication year
(MQ3) Research type
(MQ4) Contribution type
(MQ5) Empirical type
(MQ6) Dataset
(MQ7) Artifact
(MQ8) SQ model
(MQ9) SQ characteristic
Remarks

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

MQ3. A research type can be classified in the following categories [12]:

- Evaluation research: Existing SPQE approaches are implemented in practice and an evaluation of them is conducted.
- Solution proposal: An SPQE solution is proposed. This solution may be a new SPQE 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 [13], [17]:

- Data mining technique. See VI.
- Process: A series of actions, or functions leading to an SPQE result and performing operations on data.
- Method: A regular and systematic means of accomplishing SPQE.
- Tool-based technique: A technique based on a software tool to accomplish SPQE tasks.
- Model: A system representation that allows SPQE to be investigated through a hierarchical structure.
- Framework: A real or conceptual structure intended to serve as a support or guide for SPQE.
- Other, e.g. guidelines.

MQ5. The selected studies can be classified as a [18], [19]:

- Case study: An empirical inquiry that investigates an SPQE approach within its real-life context.
- Survey: A method for collecting quantitative information concerning an SPQE approach, e.g. a questionnaire.
- Experiment: An empirical method applied under controlled conditions, in order to evaluate a SPQE approach.
- History-based evaluation: Studies evaluating SPQE approaches in previously completed software projects.
- Theory: Non-empirical research approaches or theoretical evaluation of an SPQE approach.

MQ6. The data can be retrieved from [17]:

- NASA [20] is a within-company database which include 13 data sets publicly available.
- ISBSG [21] is a non-free cross-company database which includes more than 5000 software projects gathered from different countries.
- PROMISE [22] is a repository of software engineering data, which is a collection of 20 publicly available datasets and tools.
- Open source software project.
- Domain specific project. 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 [19], [24]: documentation, design module, source code or other.

MQ8. A well-known SQ model can be classified as [8]: McCall model [4], Boehm model [5], Dromey model [6], ISO/IEC 9126 standard [2], ISO/IEC 25010 standard [7], or other.

MQ9. An SQ characteristic can be classified into one of the internal and external quality characteristics proposed by ISO/IEC 9126 [2]: 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 [23], [24]. For these reasons, the ISO/IEC 9126 [2] was chosen as a reference to answer this question rather than the ISO/IEC 25010 standard [7] which appeared in 2011 and does not cover the time frame of the selected studies between 1990 and 2013.

E. 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.

III. Results

This section describes the mapping study results presented in Table IV.

A. MQ1. Publication channels

Table IV and Table V show the publication channels for SPQE research. Around 58% of the SPQE papers identified were published in journals, 21% appeared in conferences, 11% in workshops and only 5% in symposia. A book and two master thesis were also identified. The Journal of Systems and Software is a recurrent publication source for SPQE techniques, followed by the Information and Software Technology Journal. The remaining SPQE papers appeared in different journals. Note that there is no obvious conference target that attracts publication concerning SPQE approaches.

TABLE IV: SPQE result summary. Acronyms: Solution Proposal (SP), Review (Rev), Evaluation Research (ER), Data Mining (DM), History-based Evaluation (HbE), Case study (CS),Experiment (Exp), Domain Specific Project (DSP), Open Source Software (OSS), Non defined (ND), Accuracy (Acc), Functionality (Fun), Reliability (Rel), Usability (Usa), Efficiency (Eff), Maintainability (Main), Fault-prone (FP), Fault tolerance (FT).

Paper	P. Source	MQ1	MQ2	MQ3	MQ4	MQ5	MQ6	MQ7	MQ8	MQ9
[9]	JCSSE	Conference	2013	SP	DM	HbA	DSP	Doc	ISO/IEC 9126	Acc, Other
[25]	JATIT	Journal	2013	SP	DM	Exp	Other	ND	ISO/IEC 9126	All
[26]	IJFSA	Journal	2013	SP	Method	CS	DSP	Doc	ISO/IEC 9126	All
[27]	EWDTS	Symposium	2013	SP	DM	CS	DSP	Code, Design	No	No
[28]	SSRSMI	Journal	2013	SP	DM	Th	No	ND	ISO/IEC 9126	All
[29]	JUS	Journal	2013	SP	DM	Exp	Other	Code, Design, Doc	Other	Fun, Other
[30]	IJIQ	Journal	2013	ER	DM	HbA	OSS	Code	No	No
[31]	Comp. Stand. Interfaces	Journal	2013	Rev	Tool	Th	No	Code	ISO/IEC 9126	All
[32]	SIGSOFT Softw. Eng. Notes	Journal	2013	ER	Other	CS	OSS	Code	No	No
[33]	IJSEIA	Journal	2013	Rev	Method	Th	No	Code	No	No
[34]	Scientific Conf.	Conference	2013	SP	Framework	HbA	DSP	Design, Doc	All	Rel, Main, Usa, Eff, Other
[35]	ICSR	Conference	2013	SP	Method	HbA	OSS	Code	No	No
[36]	CCEM	Conference	2013	SP	DM	Th	No	ND	ISO/IEC 9126	Other
[37]	WorldCIST	Other	2013	SP	Tool	HbA	OSS	Code, Doc	No	No
[38]	SEPoW	Workshop	2013	SP	Model	Th	No	ND	No	Other
[39]	PROMISE	Conference	2013	ER	Model	HbA	OSS	Code, Doc	ISO/IEC 25010	Main, Rel
[40]	IJASA	Journal	2013	SP	Model	Th	No	ND	ISO/IEC 9126	Other
[41]	IJCA	Journal	2013	ER	Method	HbA	OSS	Code	ISO/IEC 9126, Other	No
[42]	Thesis	Other	2013	ER	Model	Survey	No	Doc	All	All
[43]	IJCSIT	Journal	2013	SP	Method	CS	DSP	ND	ISO/IEC 9126	All
[44]	IJE	Journal	2013	SP	Method	Survey	No	Code	ISO/IEC 9126	All
[45]	ACCT	Conference	2012	SP	DM	Th	No	Code, Design, Doc	No	FT
[10]	IST	Journal	2012	SP	DM	HbA	Other	Code, Design, Doc	No	No
[46]	Thesis	Other	2012	ER	Model	CS	DSP	Code	ISO/IEC 9126	Acc, Eff
[47]	IJIC	Journal	2011	SP	Framework	Th	No	Code	ISO/IEC 9126	Main
[48]	WoSQ	Workshop	2011	SP	DM	CS	OSS	Code	Other	Fun
[49]	SEW	Workshop	2011	SP	DM	HbA	OSS	Code, Design	No	Other
[50]	IWSM-MENSURA	Conference	2011	SP	Framework	CS	Other	Code, Design, Doc	Dromey model	No
[51]	WETSoM	Workshop	2011	ER	DM	HbA	OSS	Code	No	Rel
[52]	IEEE T Software Eng	Journal	2010	SP	DM	HbA	NASA	Code	No	FP
[53]	IJEST	Journal	2010	SP	Model	HbA	OSS	Code, Design	No	FP, Main
[54]	ECSA	Conference	2010	ER	Model	HbA	OSS	Code, Doc	All	Fun
[55]	Inform Sciences	Journal	2010	SP	DM	CS	DSP	Code	No	FP
[56]	IST	Journal	2010	SP	DM	HbA	NASA	Code	ISO/IEC 9126	Main
[57]	ICSE	Conference	2009	SP	Tool	Th	No	Code	No	Main
[58]	ENTC	Journal	2009	SP	Tool	HbA	OSS	Code, Design, Doc	ISO/IEC 9126	Main, Rel, Fun
[59]	ICSE	Workshop	2009	SP	DM	HbA	OSS	Code	No	No
[60]	SIGSOFT Softw. Eng. Notes	Journal	2009	SP	Process	Exp	DSP	Code	ISO/IEC 9126	All
[61]	SQJ	Journal	2008	SP	DM	CS	DSP	Design, Doc	ISO/IEC 9126	All
[62]	IEEE T Syst Man Cy A	Journal	2007	SP	DM	HbA	NASA	Code	No	FP
[63]	SCAM	Conference	2007	SP	Tool	CS	DSP	Code	No	No
[64]	HASE	Symposium	2007	ER	DM	HbA	NASA	Code	No	FP
[65]	ICICIC	Conference	2007	SP	DM	Survey	No	ND	ISO/IEC 9126	All
[66]	WoSQ	Workshop	2007	ER	Process	Exp	OSS	Code	ISO/IEC 9126	No
[67]	IST	Journal	2006	SP	DM	HbA	OSS	Doc	ISO/IEC 9126, Other	All
[68]	SNPD/SAWN	Conference	2005	SP	Model	HbA	DSP	Code, Design, Doc	ISO/IEC 9126, Other	All
[69]	IEEE Software	Journal	2004	ER	Other	Survey	No	ND	ISO/IEC 9126	All
[11]	IEEE Intell Syst	Journal	2004	ER	DM	HbA	NASA	Code, Design	No	FP
[70]	Computer	Journal	2002	SP	Process	Th	No	Code, Design, Doc	No	Rel
[71]	Fuzzy Set Syst	Journal	2002	SP	DM	HbA	DSP	Code	No	FP
[72]	JSS	Journal	2000	ER	DM	Exp	OSS	Code, Design, Doc	No	FP
[73]	JSS	Journal	1999	ER	DM	Survey	No	Code	No	Rel
[74]	JSS	Journal	1998	SP	Other	Th	No	Doc	No	No
[75]	JSS	Journal	1997	ER	DM	HbA	NASA	Code, Doc	No	FT
[76]	JSS	Journal	1997	SP	DM	HbA	DSP	Code, Design	No	FP
[77]	ISESS	Symposium	1995	SP	Framework	Th	No	ND	Bohem model	Other
[78]	ITOR	Journal	1994	ER	Other	HbA	DSP	Doc	No	No

TABLE V: Publication channel

Publication channel	Total
Journal	33
Conference	12
Workshop	6
Symposium	3
Other	3
Total	57

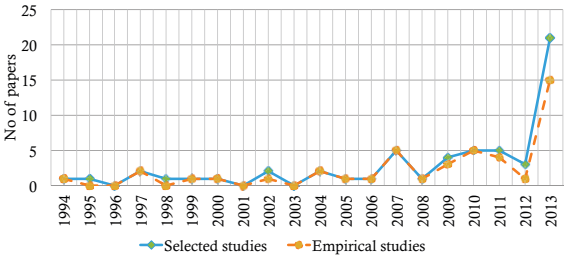

Fig. 2: Publication per year

B. MQ2. Publication trend

Fig. 2 shows the publication trends of the selected papers. This figure shows also the trend of empirical studies identified in this study. The trend of SPQE publications that correspond to our search string is characterized by discontinuity. The interest in SPQE has increased in the last

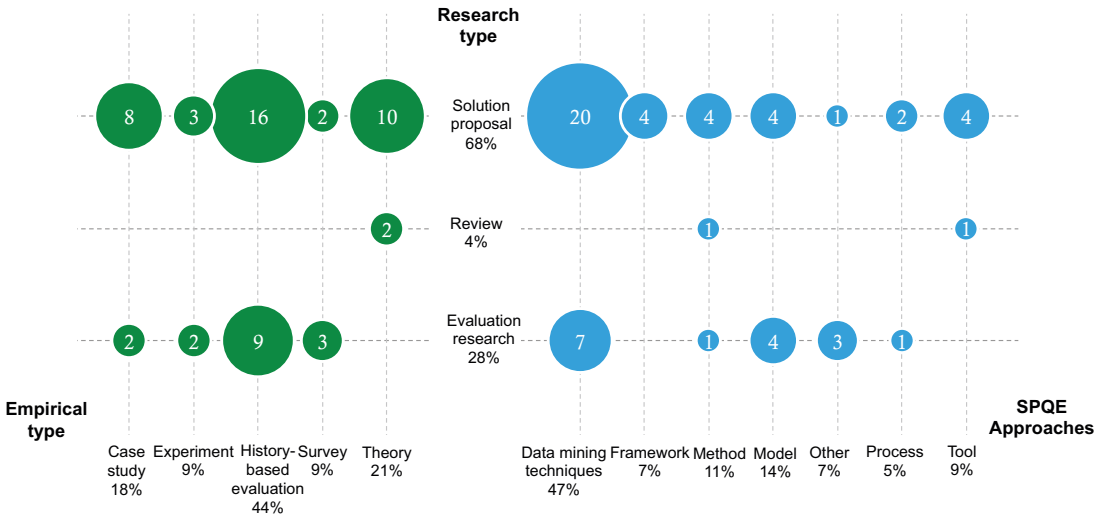

Fig. 3: Research types, SPQE approaches and empirical types

years to reach a peak in 2013. Notice that the empirical trend is almost parallel to SPQE trend in many time slots between 1994 and 2013, which indicates that the majority of SPQE research is empirically validated.

C. MQ3. Research types

Fig. 3 shows the research type of the selected papers. Around 68% of the selected papers were solution proposal studies and except two reviews, the rest of papers were studies that were undertaken to evaluate SPQE existing approaches. This result shows that the main concern of researchers in the SPQE domain is to propose and develop approaches to enhance SPQE. Fig. 3 shows also that 82% of solution proposals were empirically validated and that 32% of the suggested solutions are data mining techniques.

D. MQ4. SPQE approaches

Fig. 3 presents the SPQE approaches extracted from the selected papers. The approaches most frequently reported are those of data mining, principally, as shown in Fig. 4, fuzzy logic [25], [28], [29], [36], [61], [65], [67], [71], regression [51], [72], [73], [76], clustering [11], [45], [62] and pattern matching techniques [10], [49], [59]. Models, methods, tools, frameworks and processes were identified in the selected studies. Other techniques in this study were also identified, such as a SPQE technique using Cronbach's alpha coefficient [69] and trade-off analysis [74].

E. MQ5. Empirical types

Fig. 3 also shows if the selected studies were empirically validated and presents the empirical types used in the validation of the selected papers. Only 21% of the selected studies were not evaluated empirically. 44% of the selected papers used existing data sets to evaluate SPQE approaches, 18% were evaluated through case studies while the rest were evaluated with experiments or by using surveys.

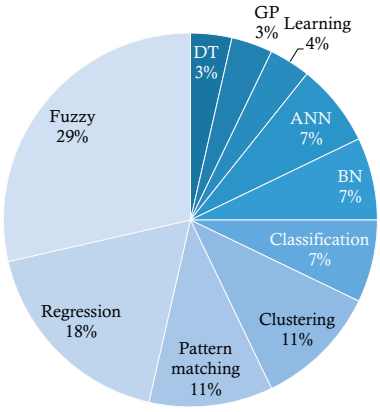

Fig. 4: Data mining techniques in SPQE selected studies (27 studies)

F. MQ6. Data sets

Fig. 5 shows the data sets used to validate SPQE approaches. The data sets that were used in SPQE papers were mainly retrieved from open source software projects or domain specific projects or NASA. PROMISE and ISBSG data sets were not used to validate SPQE approaches. Few papers have used other data sets which were based on simulation data.

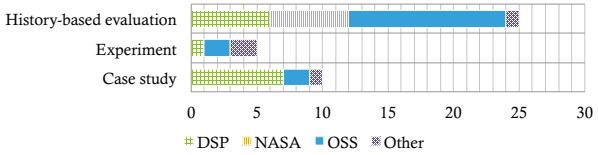

Fig. 5: Data sets used in the validation of SPQE approaches (40 studies)

G. MQ7. Artifacts

Fig. 6 presents the software artifacts identified in the selected studies. This figure shows whether the artifact has been mentioned exclusively in a paper or it has been mentioned with other artifacts in the same paper. Source code was discussed in 70% of the selected studies. 40% of the selected studies focussed exclusively on source code. 35% of the selected studies discussed documentation, mainly requirements documentations such as in [26]. Design module artifact was reported by 26% of SPQE research in only papers reporting more than an artifact.

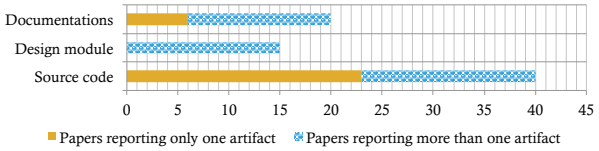

Fig. 6: Artifacts (48 studies)

H. MQ8. SQ models

The results shown in Fig. 7 reveal that around 50% of SPQE papers do not cite any well-known SQ model. The principal model cited in the selected studies was the ISO/IEC 9126 standard [2]. McCall [4], Dromey [6] and Boehm [5] were also cited. Note that only one paper [79] has cited ISO/IEC 25010 [7]. Another SQ model that was mentioned by Wagner [48] was the Quamoco quality model, which is a quality model with a corresponding quality assessment method whose objective is to facilitate continuous improvement based on objective and quantitative feedback and whose origins lie in the quality improvement paradigm and the Goal/Question/Metric GQM approach. Some papers cited different models and standards that helped them in the design of SPQE techniques, such as: ISO/IEC 14598 for software product evaluation, the Capability Maturity Model (CMM), ISO/IEC 15504 for process assessment, and ISO 9000 for quality management systems.

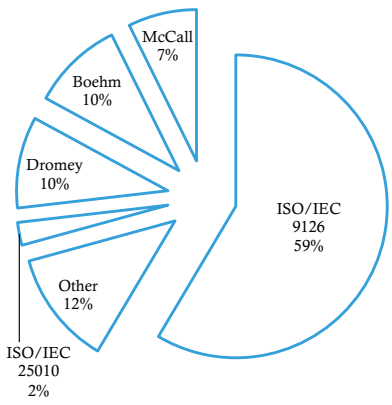

Fig. 7: SQ well-known model (29 studies)

I. MQ9. SQ characteristics

25% of SPQE papers did not mention any SQ characteristics. Table IV shows that the main characteristic used in the evaluation of SQ in literature is reliability including fault-proneness. followed by maintainability and functionality. 23% of the selected studies had used all the ISO/IEC 9126 characteristics to evaluate SPQ. Other characteristics were used by some papers such as reusability, availability, flexibility and extendibility. All the characteristics presented in Table IV, with the exception of fault-proneness, are actually part of the internal and external sub-characteristics of the ISO/IEC 9126 standard [2]. 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 [2].

IV. Summary and Discussion

A. Principal findings

- The amount of publication in recognized and stable journals and conferences indicates that SPQE is taken very seriously by researchers.
- The interest in evaluating software product quality has increased dramatically last year because more methods and data mining techniques were proposed. 2014 will prove if 2013 is an exemption or a starting point for more SPQE research.
- The most frequent research type identified for SPQE was that of solution proposal. This result indicates that researchers are still searching for good approaches with which to enhance SPQE. Few researchers chose to evaluate existing SPQE techniques.
- The majority of SPQE approaches that were found are data mining techniques, principally fuzzy logic, regression and clustering techniques, which is quite normal as there is an increasing interest in using data mining techniques in many software engineering subfields [80] and in other domains.
- The majority of the selected studies were evaluated empirically which indicates that, in an empirical sense, the research is in a good state. The selected papers have principally used data from open source projects, domain specific projects or from existing data sets such as NASA to evaluate SPQE solutions. Case studies were used to evaluate SPQE approaches and a few experiments were conducted in the SPQE papers selected. Empirical validation using surveys were also conducted in some selected papers.
- Source code was the main artifacts concerned by SPQE approaches, which indicates that researchers are more interested to evaluate internal SQ. Documentations and design modules were also mentioned in some of the selected studies.
- Half of the selected studies based their solutions on SQ models, particularly the ISO/IEC 9126 standard [2]. The ISO/IEC 25010 standard [7] appeared in 2011 to replace the ISO/IEC 9126 [2]. But of the 15 selected

papers which were published after 2011 and which cited SQ models, only one paper cited the ISO/IEC 25010 standard [7]. Authors may include this standard more frequently in their publications, while in the coming years the ISO/IEC 9126 [2] may gradually disappear from SQ literature [81]. 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 new characteristic in ISO/IEC 25010 and quality in use has five characteristics instead of the four characteristics of ISO/IEC 9126.

- 30% of the selected studies have not mentioned any well-known SQ models, however they have evaluated 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, are difficult to apply and require a lot of adaptation to particular situations [50] as they provide only the general SQ framework. Reliability was the most frequently reported SQ characteristic in SPQE literature, and was in most cases achieved using fault-proneness. Since according to Lyu et al. [82] the cost of software application failures is growing and the failures are increasingly impacting on business performance, it is critical to ensure that all modules are error free [83], [84]. 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 [85].

B. Implications of the results

The findings of our systematic mapping study have implications for researchers and practitioners who are working in the SQ domain, since this study will allow them to discover the existing SPQE approaches and techniques in the literature. Moreover, the empirical studies presented 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 [7]. The evaluation of SQ 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 datasets in software engineering that the ones mentioned in this paper, such as FLOSSmole [86], FLOSSMetrics [87], and SECOLD [88] which is the first online Software ECOsystem Linked Data platform of source code facts [89].

C. Threats to validity

- *Construct validity:* Construct threats to validity in a mapping study are related to the identification of primary studies [15], [90]. A high quality mapping study should be based on a stringent search process [91]. In order to ensure that as many relevant primary studies as possible were being included, different terms for SPQE techniques were included in the search string. The fact that 19 out of the 25 search terms are related to data mining and artificial intelligence do not affect the results because they are substitutes of both words. Other terms that refer to evaluation were also included in the search string in order to obtain a result set that was as complete as possible. 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 [92]. 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 [15], [90]. 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 judgement 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. Data extraction from prose could also result in a misclassification, but this problem was addressed by developing a classification scheme on the basis of widely accepted guidelines [2] and terminology proposed for use in SPQE. This would, therefore, only have a minor influence on the general classification derived in this study.
- *Conclusion validity:* In the case of a mapping study, this threat refers to factors such as missing studies and incorrect data extraction [90]. The aim is to control these factors so that a systematic mapping study can be performed by other researchers [15], [92], [93] who will draw the same conclusions [94]. Bias both as regards selecting and classifying primary studies and analyzing data may therefore affect the interpretation of the results. 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 data extracted 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 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 [93], [95], [96]. The

systematic mapping results were considered with regard to the SPQE domain, and the validity of the conclusions drawn in this paper concerns only the SPQE 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.

V. Conclusions and future work

The overall goal of this study is to summarize the existing knowledge as regards SPQE approaches. Papers dealing with SPQE approaches from between 1990 and 2013 were identified. 57 papers were selected. The main publication sources of the papers identified were journals. The result of this study shows that the approaches used to enhance SPQE that are most frequently reported in literature are those of data mining, and principally fuzzy logic and regression techniques. Another interesting result is that half of the selected papers mention at least one well-known SQ model in their research. More attention should be paid to the ISO/IEC 25010 standard [7] in the design of SPQE approaches. Reliability is the principal SQ characteristic through which the papers selected evaluate the overall SQ, 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 SPQE, which was deduced from the fact that solution proposals were identified more frequently than evaluation research. Source code was the main artifact concerned by SPQE research. The majority of the selected papers were history-based evaluations using existing data sets. The data sets were mainly obtained from open source software projects and domain specific projects. Only a few papers extracted data from repositories such as NASA.

This study could help practitioners to identify approaches with which to enhance the SPQE 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 SPQE research results. Ongoing research is based on proposing an empirical method with which to evaluate SPQ.

APPENDIX

VI. An overview of data mining techniques

- Regression analysis [97]: a statistical technique for estimating the relationships among variables. It includes many techniques for modelling and analysing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables.
- Clustering [98]: a method of unsupervised learning and a common technique for statistical data analysis used in many fields, including machine learning , data mining, pattern recognition, image analysis and bioinformatics. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to some predesignated criterion or criteria, while observations drawn from different clusters are dissimilar.
- Bayesian network (BN) [99]: a relationship network that uses statistical methods to represent probability relationships between different elements. Its theoretical foundation is the Bayes rule [100].
- Artificial neural networks (ANN) [101]: learning systems inspired by organic neural systems, which are known to be very successful when learning to solve problems. They comprise a network of simple interconnected units called neurons. The connections between the neurons are weighted and the values of these weights determine the function of the network.
- Fuzzy logic [102]: In this logic, the underlying logic is approximate rather than exact as everything is a matter of degree. So when A is a Fuzzy set and x is a relevant object, the proposition that x is a member of A is not necessarily either true or false. The degree to which x is actually a member of A may only be true to a certain degree.
- Learning technique [103]: the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function.
- Genetic algorithms (GA) [104]: adaptive heuristic search methods that may be used to solve all kinds of complex search and optimization problems, for which no efficient heuristic method has been developed. They are based on the evolutionary ideas of natural selection and genetic processes of biological organisms.
- Genetic programming (GP) [105]: an extension of GA. It is a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task.
- Case-based reasoning (CBR) [106]: a technique that aims to find solutions to new problems based on past experiences, which are represented by cases in a case library. The case library and the associated retrieval and decision rules constitute a CBR model. In the context of a classification problem, each case in the case library has known attributes and class membership.
- Rough Sets (RS) [107]: a formal approximation of a crisp set (i.e., conventional set) in terms of a pair of sets which give the lower and the upper approximation of the original set. In the standard version of rough set theory, the lower- and upper-approximation sets are crisp sets, but in other variations, the approximating sets may be fuzzy sets.
- Decision tree (DT) [108]: this represents a map of the reasoning process. A decision tree can be used to explain why a question is being asked. It is a formalism for expressing the mapping from attribute values to classes.

- Classification [109]: this maps data into predefined groups or classed. Classification algorithms require that the classes be defined based on data attribute values.
- Pattern Matching [109]: or pattern recognition finds occurrences of predefined pattern in data. It is used in many diverse applications. A text editor uses pattern matching to find occurrences of a string in the text being edited.

ACKNOWLEDGMENTS

This research is part of the project Software Project Management using DataMining Techniques, (AP2010-2013), financed by Mohammed V Souissi 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] B. Kitchenham and S. L. Pfleeger, "Software quality: The elusive target," *IEEE Softw.*, vol. 13, no. 1, pp. 12–21, Jan. 1996.

[2] *ISO/IEC Standard. ISO-9126 Software Product Evaluation - Quality Characteristics and Guidelines for Their Use*, ISO/IEC Std., 1999.

[3] (2009) Standish-Group, CHAOS summary. [Online]. Available: http://blog.standishgroup.com/pmresearch

[4] J. A. McCall, *Quality Factors*. John Wiley & Sons, Inc., 2002.

[5] B. W. Boehm, J. R. Brown, H. Kaspar, and M. Lipow, *Characteristics of software quality*, ser. TRW Softw. Technol. Amsterdam: North-Holland, 1978.

[6] R. G. Dromey, "Cornering the Chimera," *IEEE Softw.*, vol. 13, no. 1, pp. 33–43, 1996.

[7] *ISO/IEC 25010 standard. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.*, ISO/IEC Std., 2011.

[8] M. Ortega, M. Pérez, and T. Rojas, "Construction of a systemic quality model for evaluating a software product," *Software Quality Control*, vol. 11, no. 3, pp. 219–242, Jul. 2003.

[9] W. Leopairote, A. Surarerks, and N. Prompoon, "Evaluating software quality in use using user reviews mining," in *Computer Science and Software Engineering (JCSSE), 2013 10th International Joint Conference on*. IEEE, 2013, pp. 257–262.

[10] A. Ampatzoglou, G. Frantzeskou, and I. Stamelos, "A methodology to assess the impact of design patterns on software quality," *Inf. Softw. Technol.*, vol. 54, no. 4, pp. 331–346, Apr. 2012.

[11] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, "Analyzing software measurement data with clustering techniques," *IEEE Intelligent Systems*, vol. 19, no. 2, pp. 20–27, Mar. 2004.

[12] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

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

[14] L. M. Karg, M. Grottke, and A. Beckhaus, "A systematic literature review of software quality cost research," *Journal of Systems and Software*, vol. 84, no. 3, pp. 415–427, Mar. 2011.

[15] F. Elberzhager, J. Münch, and V. T. N. Nha, "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, Jan. 2012.

[16] J. Landis and G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, pp. 159–174, 1977.

[17] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012.

[18] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[19] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, "Requirements engineering education: a systematic mapping study," *Requirements Engineering*, pp. 1–20, 2013.

[20] (2013) NASA Metrics Data Program (MDP) Repository. [Online]. Available: http://mdp.ivv.nasa.gov/

[21] ISBSG, "International Software Benchmarking Standards Group," 2013. [Online]. Available: http://www.isbsg.org/

[22] T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan, "The PROMISE Repository of empirical software engineering data," 2012. [Online]. Available: http://promisedata.googlecode.com

[23] "Guide to the Software Quality Body of Knowledge (SQuBOK)," JUSE: The Union of Japanese Scientists and Engineers, Tech. Rep., 2007.

[24] A. Abran and J. W. a. Moore, *Guide to the software engineering body of knowledge (SWEBOK)*, C. Los Alamitos, Ed. IEEE Computer Society, 2004. [Online]. Available: http://www.swebok.org/

[25] X. GENG, L. YANG, and X. HUANG, "A software quality evaluation method based on the principle of maximum coordination and subordination." *Journal of Theoretical & Applied Information Technology*, vol. 47, no. 1, 2013.

[26] V. Pasrija and P. R. Srivastava, "Evaluation of software quality using choquet integral approach," *International Journal of Fuzzy System Applications (IJFSA)*, vol. 3, no. 1, pp. 51–81, 2013.

[27] O. Pomorova and T. Hovorushchenko, "Artificial neural network for software quality evaluation based on the metric analysis," in *East-West Design & Test Symposium, 2013*. IEEE, 2013, pp. 1–4.

[28] M. HAJDU-MACELARU, "Software quality evaluation using fuzzy theory1," *Scientific Studies & Research. Series Mathematics & Informatics*, vol. 23, no. 1, 2013.

[29] X. Wang, M. Ceberio, S. Virani, A. Garcia, and J. Cummins, "A hybrid algorithm to extract fuzzy measures for software quality assessment," *Journal of Uncertain Systems*, vol. 7, no. 3, pp. 219–237, 2013.

[30] L. Yu and S. Ramaswamy, "Using bug report as a software quality measure: an empirical study," *International Journal of Information Quality*, vol. 3, no. 2, pp. 164–183, 2013.

[31] P. Tomas, M. Escalona, and M. Mejias, "Open source tools for measuring the internal quality of java software products. a survey," *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 244–255, 2013.

[32] G. Singh, "Metrics for measuring the quality of object-oriented software," *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 5, pp. 1–5, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2507288.2507311

[33] M.-C. Lee and T. Chang, "Software measurement and software metrics in software quality," *International Journal of Software Engineering & Its Applications*, vol. 7, no. 4, 2013.

[34] C. Otieno, W. Mwangi, and S. Kimani, "Framework to assess software quality in erp systems," in *Scientific Conference Proceedings*, 2013.

[35] Y. Zou, C. Liu, Y. Jin, and B. Xie, "Assessing software quality through web comment search and analysis," in *Safe and Secure Software Reuse*. Springer, 2013, pp. 208–223.

[36] N. Baliyan and S. Kumar, "Quality assessment of software as a service on cloud using fuzzy logic," in *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 2013, pp. 1–6.

[37] N. R. Carvalho, A. Simões, and J. J. Almeida, "Open source software documentation mining for quality assessment," in *Advances in Information Systems and Technologies*. Springer, 2013, pp. 785–794.

[38] S. Rehman and S. U. Khan, "Software quality assessment model for global software development," in *e-Proceeding of Software Engineering Postgraduates Workshop (SEPoW)*, 2013, p. 104.

[39] K. Lochmann, J. Ramadani, and S. Wagner, "Are comprehensive quality models necessary for evaluating software quality?" in *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*. ACM, 2013, p. 3.

[40] T. A. Alrawashdeh, M. Muhairat, and A. Althunibat, "Evaluating the quality of software in erp systems using the iso 9126 model," *International Journal of Ambient Systems and Applications (IJASA)*, 2013.

[41] P. Kumar, R. Yadav, and D. Bansal, "Usability and evaluation of software quality using software metrics," *International Journal of Computer Applications*, vol. 71, no. 3, pp. 9–14, 2013.

[42] D. Gade, "The evaluation of software quality," Master's thesis, University of Nebraska - Lincoln, 2013.

[43] N. F. Duarte Filho, C. I. da Silva e Pádua, P. H. de Souza Bermejo, A. L. Zambalde, U. S. de Barros *et al.*, "Saasquality-a method for quality evaluation of software as a service (saas)." *International Journal of Computer Science & Information Technology*, vol. 5, no. 3, 2013.

[44] B. Jayasingh, S. D. Virivinti, and N. Pranav, "Expert rating based software quality evaluation," *International Journal of Engineering*, vol. 2, no. 10, 2013.

[45] P. Dhiman, Manish, and R. Chawla, "A clustered approach to analyze the software quality using software defects," *Advanced Computing & Communication Technologies, International Conference on*, pp. 36–40, 2012.

[46] S. HABIBI, "Evaluating the quality of a software application for web services integration," Master's thesis, University of Gothenburg, 2012.

[47] J. G and C. V. Babu, "An integrated approach for measuring software quality and code readability," vol. 1, no. 2, 2011.

[48] S. Wagner, "The use of application scanners in software product quality assessment," in *Proceedings of the 8th international workshop on Software quality*, ser. WoSQ '11. New York, NY, USA: ACM, 2011, pp. 42–49.

[49] F. A. Fontana and S. Maggioni, "Metrics and antipatterns for software quality evaluation," in *34th IEEE Software Engineering Workshop*, 2011, pp. 48–56.

[50] B. Vanderose and N. Habra, "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*, pp. 263–268, 2011.

[51] I. Turnu, G. Concas, M. Marchesi, and R. Tonelli, "The fractal dimension metric and its use to assess object-oriented software quality," in *Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics*, ser. WETSoM '11. New York, NY, USA: ACM, 2011, pp. 69–74.

[52] Y. Liu, T. M. Khoshgoftaar, and N. Seliya, "Evolutionary optimization of software quality modeling with multiple repositories," *IEEE Trans. Softw. Eng.*, vol. 36, no. 6, pp. 852–864, Nov. 2010.

[53] M. V. V. Saradhi and B. R. Sastry, "ESPQ: A new object oriented design metric for software quality measurement," *International Journal of Engineering Science and Technology*, 2010.

[54] A. K. Groven, K. Haaland, R. Glott, and A. Tannenberg, "Security measurements within the framework of quality assessment models for free/libre open source software," in *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ser. ECSA '10. New York, NY, USA: ACM, 2010, pp. 229–235.

[55] T. M. Khoshgoftaar, Y. Xiao, and K. Gao, "Software quality assessment using a multi-strategy classifier," *Information Sciences*, p. in press, 2010.

[56] S. Wagner, "A bayesian network approach to assess and predict software quality using activity-based quality models," *Information and Software Technology*, vol. 52, no. 11, pp. 1230–1241, 2010.

[57] G. Gousios and D. Spinellis, "Alitheia core: An extensible software quality monitoring platform," in *Proceedings of the 31st International Conference on Software Engineering*, ser. ICSE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 579–582.

[58] D. Spinellis, G. Gousios, V. Karakoidas, P. Louridas, P. J. Adams, I. Samoladas, and I. Stamelos, "Evaluating the quality of open source software," *Electron. Notes Theor. Comput. Sci.*, vol. 233, pp. 5–28, Mar. 2009.

[59] M. R. Marri, S. Thummalapenta, and T. Xie, "Improving software quality via code searching and mining," in *Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*, ser. SUITE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 33–36.

[60] A. Kumar, P. S. Grover, and R. Kumar, "A quantitative evaluation of aspect-oriented software quality model (aosquamo)," *SIGSOFT Softw. Eng. Notes*, vol. 34, no. 5, pp. 1–9, Oct. 2009.

[61] C.-W. Chang, C.-R. Wu, and H.-L. Lin, "Integrating fuzzy theory and hierarchy concepts to evaluate software quality," *Software Quality Journal*, vol. 16, no. 2, pp. 263–276, 2008.

[62] N. Seliya and T. M. Khoshgoftaar, "Software quality analysis of unlabeled program modules with semisupervised clustering," *Trans. Sys. Man Cyber. Part A*, vol. 37, no. 2, pp. 201–211, Mar. 2007.

[63] S. Roubtsov, A. Telea, and D. Holten, "Squavisit: A software quality assessment and visualisation toolset," in *Proceedings of the Seventh IEEE International Working Conference on Source Code Analysis and Manipulation*, ser. SCAM '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 155–156.

[64] M. Shin, S. Ratanothayanon, A. L. Goel, and R. A. Paul, "Parsimonious classifiers for software quality assessment," in *Proceedings of the 10th IEEE High Assurance Systems Engineering Symposium*, ser. HASE '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 411–412.

[65] L. Lin and H.-M. Lee, "A fuzzy software quality assessment model to evaluate user satisfaction," in *Proceedings of the Second International Conference on Innovative Computing, Informatio and Control*, ser. ICICIC '07. Washington, DC, USA: IEEE Computer Society, 2007, p. 438.

[66] K. Stroggylos and D. Spinellis, "Refactoring–does it improve software quality?" in *Proceedings of the 5th International Workshop on Software Quality*, ser. WoSQ '07. Washington, DC, USA: IEEE Computer Society, 2007, p. 10.

[67] C.-B. Chen, C.-T. Lin, C.-H. Wang, and C.-W. Chang, "Model for measuring quality of software in DVRS using the gap concept and fuzzy schemes with ga," *Inf. Softw. Technol.*, vol. 48, no. 3, pp. 187–203, Mar. 2006.

[68] K. Lee and S. J. Lee, "A quantitative software quality evaluation model for the artifacts of component based development," in *Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks*, ser. SNPD-SAWN '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 20–25.

[69] H.-W. Jung, S.-G. Kim, and C.-S. Chung, "Measuring software product quality: A survey of ISO/IEC 9126," *IEEE Softw.*, vol. 21, no. 5, pp. 88–92, Sep. 2004.

[70] N. F. Schneidewind, "Body of knowledge for software quality measurement," *Computer*, vol. 35, no. 2, pp. 77–83, Feb. 2002.

[71] S. S. So, S. D. Cha, and Y. R. Kwon, "Empirical evaluation of a fuzzy logic-based software quality prediction model," *Fuzzy Sets Syst.*, vol. 127, no. 2, pp. 199–208, Apr. 2002.

[72] L. C. Briand, J. Wüst, J. W. Daly, and D. V. Porter, "Exploring the relationship between design measures and software quality in object-oriented systems," *J. Syst. Softw.*, vol. 51, no. 3, pp. 245–273, May 2000.

[73] E. J. Weyuker, "Evaluation techniques for improving the quality of very large software systems in a cost-effective way," *J. Syst. Softw.*, vol. 47, no. 2-3, pp. 97–103, Jul. 1999.

[74] X. F. Liu, "A quantitative approach for assessing the priorities of software quality requirements," *J. Syst. Softw.*, vol. 42, no. 2, pp. 105–113, Aug. 1998.

[75] F. Lanubile and G. Visaggio, "Evaluating predictive quality models derived from software measures: lessons learned," *J. Syst. Softw.*, vol. 38, no. 3, pp. 225–234, Sep. 1997.

[76] R. Takahashi, "Software quality classification model based on McCabe's complexity measures," *J. Syst. Softw.*, vol. 38, no. 1, pp. 61–69, Jul. 1997.

[77] R. Gulezian, "Software quality measurement and modeling, maturity, control and improvement," in *Proceedings of the 2nd IEEE Software Engineering Standards Symposium*, ser. ISESS '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 52–59.

[78] C. Maroto and P. Tormos, "Project management: an evaluation of software quality," vol. 1, no. 2, pp. 209–221, 1994.

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

[80] T. Xie, S. Thummalapenta, D. Lo, and C. Liu, "Data mining for software engineering," *Computer*, vol. 42, no. 8, pp. 55–62, Aug. 2009.

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

[82] M. R. Lyu, "Software reliability engineering: A roadmap," in *2007*

Future of Software Engineering, ser. FOSE '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 153–170.

[83] C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, "Improving software-quality predictions with data sampling and boosting," *Trans. Sys. Man Cyber. Part A*, vol. 39, no. 6, pp. 1283–1294, Nov. 2009.

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

[85] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, "Unsupervised learning for expert-based software quality estimation," in *Proceedings of the Eighth IEEE international conference on High assurance systems engineering*, ser. HASE'04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 149–155.

[86] J. HOWISON, M. CONKLIN, and K. CROWSTON, "FLOSSmole: A collaborative repository for FLOSS research data and analyses," vol. 1, no. 3, pp. 17–26, 2006. [Online]. Available: http://flossmole.org/

[87] I. Herraiz, D. Izquierdo-Cortazar, F. Rivas-Hernández, J. Gonzalez-Barahona, G. Robles, S. Duenas-Dominguez, C. Garcia-Campos, J. F. Gato, and L. Tovar, "Flossmetrics: Free/libre/open source software metrics," in *13th European Conference on Software Maintenance and Reengineering, 2009. CSMR'09.* IEEE, 2009, pp. 281–284. [Online]. Available: http://flossmetrics.org/

[88] I. Keivanloo, C. Forbes, J. Rilling, and P. Charland, "Towards sharing source code facts using linked data," in *Proceedings of the 3rd International Workshop on Search-Driven Development: Users, Infrastructure, Tools, and Evaluation*, ser. SUITE '11. New York, NY, USA: ACM, 2011, pp. 25–28. [Online]. Available: http://www.secold.org/

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

[90] A. Ampatzoglou, S. Charalampidou, and I. Stamelos, "Research state of the art on GoF design patterns: A mapping study," *Journal of Systems and Software*, 2013.

[91] B. A. Kitchenham, D. Budgen, and O. Pearl Brereton, "Using mapping studies as the basis for further research–a participant-observer case study," *Information and Software Technology*, vol. 53, no. 6, pp. 638–651, 2011.

[92] V. Garousi, A. Mesbah, A. Betin-Can, and S. Mirshokraie, "A systematic mapping study of web application testing," *Information and Software Technology*, vol. 55, no. 8, pp. 1374–1396, 2013.

[93] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 285–311.

[94] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, and S. Beecham, "Tools used in global software engineering: A systematic mapping review," *Inf. Softw. Technol.*, vol. 54, no. 7, pp. 663–685, Jul. 2012.

[95] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering: an introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[96] P. R. Mateo, M. P. Usaola, and J. L. Fernández Alemán, "Validating 2nd-order mutation at system level," *IEEE Transactions on Software Engineering*, vol. 39, pp. 570–587, 2012.

[97] N. R. Draper and H. Smith, *Applied Regression Analysis*. New York, USA : Wiley, 1998.

[98] J. A. Hartigan, *Clustering Algorithms*, 99th ed. New York, NY, USA: John Wiley & Sons, Inc., 1975.

[99] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, ser. WI '03. Washington, DC, USA: IEEE Computer Society, 2003, p. 372.

[100] D. Heckerman, "A tutorial on learning with bayesian networks," in *Innovations in Bayesian Networks*, ser. Studies in Computational Intelligence, D. Holmes and L. Jain, Eds. Springer Berlin Heidelberg, 2008, vol. 156, pp. 33–82.

[101] B. Yegnanarayana, *Artificial Neural Networks*. Prentice-Hall of India Pvt.Ltd, 2004.

[102] J. Yen and R. Langari, *Fuzzy logic: intelligence, control, and information*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[103] V. N. Vapnik, "An overview of statistical learning theory," *Trans. Neur. Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[104] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. Addison-Wesley Professional, 1989.

[105] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[106] D. B. Leake, "Case-based reasoning," in *Encyclopedia of Computer Science*. Chichester, UK: John Wiley and Sons Ltd., 2003, pp. 196–197.

[107] Z. Pawlak, "Rough sets," in *Proceedings of the 1995 ACM 23rd annual conference on Computer science*, ser. CSC '95. New York, NY, USA: ACM, 1995, pp. 262–264.

[108] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[109] H. D. Margaret, "Data mining introductory and advanced topics," *Pearsons Education Inc*, 2003.