

Identifying Relevant Product Quality Characteristics in the Context of Very Small Organizations

Gabriel Alberto García-Mireles

Departamento de Matemáticas, Universidad de Sonora
Blvrd. Encinas y Rosales s/n, 83000 Hermosillo, Sonora, México
mireles@mat.uson.mx

Abstract. Software process improvement models have been developed to support very small organizations in achieving the software quality expected by their customers. Software quality, however, allows to be studied from process and product perspectives. The aim of this work is to attain insights into the extent to which the product quality view is addressed in very small organizations. A mapping method was therefore applied in order to identify whether the product quality characteristics described in ISO/IEC 25010 are included in the ISO/IEC 29110 basic profile, a process model targeted towards very small organizations. In addition, an empirical study was conducted to discover the quality characteristics that practitioners at very small organizations consider relevant. As a result, it was found that practices in ISO/IEC 29110 are consistent as regards functional correctness quality subcharacteristic, although practitioners are also considering additional product quality characteristics.

Keywords: software quality, very small organizations, software process, ISO/IEC 29110, ISO/IEC 25010.

1. Introduction

Software process improvement models provide companies with support that helps them achieve their business goals as regards delivering quality products. Although large organizations have reported success when using these models, implementing management practices in order to support software development activities is a challenge for small and very small companies [1]. A very small entity (VSE) is defined as an “entity (enterprise, organization, department or project) having up to 25 people” [2]. The characteristics of VSEs make the implementation of traditional software process improvement models such as CMMI and ISO/IEC 15504 difficult [1].

Small software companies have different requirements when implementing process models [3]. The main factor that makes the implementation of process models difficult is the lack of resources in terms of the number of employees, budget and time [2]. These companies are extremely flexible and have a flat structure that allows innovation [4]. They usually focus on a market niche which is not considered by large companies and also build components for others companies, or offer services or maintenance of products [4]. They do not have enough staff to provide their software engineering roles with functional expertise, and their tight financial constraints prevent them from

attaining the appropriate training, carrying out process improvement and efficiently responding to risks [3-5]. With regard to software processes, the development life cycle is often highly simplified, and activities such as analysis, software construction and testing are often not formalized [3, 5].

Despite their lack of resources, very small companies represent a significant force in several economical regions and countries. Basri et al. [3] reported that companies with 1 to 10 employees represent 85% of the Information Technology sector. A survey carried out in Montreal found that 78% of software development enterprises have less than 25 employees [3]. In Mexico, most of the companies (99.8%) have between 1 to 130 employees [6].

Indeed, the global software industry dominant force relies on small and very small organizations [1]. A number of VSEs provide software components for the manufacturing sector. Defects in software components may, however, increase production costs in this industrial sector [7], and VSEs, therefore, have a pressing need to develop their products efficiently, effectively and with high quality [3].

The characteristics of small companies have inspired the development of several process models with which to support their improvement, such as Competisoft and Moprosoft [8]. In the context of VSEs, the ISO/IEC 29110 was developed to assess and improve the software lifecycle processes of VSEs. It was specifically developed to improve product and/or service quality and process performance [2] and is becoming a widely adopted standard [1].

Bearing in mind that one of the purposes of ISO/IEC 29110 is to improve product quality, we focused on this aspect in order to investigate the extent to which software product quality is addressed. Software quality is difficult to define and can be studied by considering different perspectives. This work focuses on two of them: the process view and the product view. The process view of quality is based on the idea that a mature process can increase the quality of a software product, and, as a result, software organizations implement development and management practices that enhance software product quality. Under this view, defect counts are considered surrogates with which to measure software quality. This view is the most dominant in the software process improvement area and includes using models such as CMMI, ISO/IEC 15504, and ISO/IEC 12207.

The product quality view, on the other hand, focuses on the characteristics of software that is under development to achieve customers' and users' needs [9]. It may employ software quality models such as ISO/IEC 25010 [10] to determine quality requirements, measure product quality and evaluate software quality. As regards to the level of quality that an organization should address in software, software organizations might address a holistic quality view where both process view and product view are taken into account [9].

Since enhancing product quality is a business strategy that a VSE can implement, the purpose of this work is to explore the extent to which practices recommended to improve process also contribute to increase specific quality product attributes. A first step is to determine which quality product characteristics are addressed in the ISO/IEC 29110 Basic Profile [11]. ISO/IEC 25010 was used to identify relevant quality characteristics by applying a mapping method between both standards. Mappings are effective in identifying requirements from different quality models and supporting the assessment of multiple models [12]. A second step is to conduct an empirical study, in

the form of an exploratory survey, to identify those quality characteristics that are relevant to practitioners. The purpose of the study was to attain insights into practitioners' perceptions of relevant product quality characteristics that they need to address when working in a VSE. The most direct means to identify quality characteristics that VSEs address in their projects is to request quality requirements, since they can be categorized into the quality subcharacteristics of the ISO/IEC 25010. A questionnaire was, consequently, created in order to identify relevant quality characteristics.

This article is an extended version of a paper presented in the International Conference on Software Process Improvement (CIMPS 2015) [29]. In comparison with the conference paper, this extended version describes the design and results of an exploratory survey conducted to get insights as regards the specific quality characteristics addressed by software developers in VSEs. In addition, the Related Work section overviews approaches to create questionnaires that rely on the product quality model.

The rest of the paper is organized as follows: Section 2 describes related work, and Section 3 provides a description of the mapping between ISO/IEC 29110 and ISO/IEC 25010. Section 4 describes the exploratory survey, while Section 5 presents a discussion. Finally, conclusions and future work are presented in Section 6.

2. Related Work

This Section is composed of two parts. The first one describes the main published works about the mapping of quality models of both quality views: process and product. The second part reviews approaches in order to identify relevant quality requirements in software projects as a means to discover relevant software quality characteristics. The reviewed methods provide the basis to develop a questionnaire to use with practitioners working in VSEs.

2.1. Product and Process Mappings

Quality approaches can be categorized based on the main artifacts targeted: product or process. It is recognized that software organizations must deal with several quality models simultaneously [12, 13]. However, several issues can arise when an organization implements them [14]. The differences among quality models concerning their structure, granularity, and vocabulary can erode the expected benefits from an improvement initiative [12]. Furthermore, most of the mappings on quality models have been carried out with process models [12, 15]. Few reports have described how to integrate both process and product quality views.

Several mappings between process and product quality standards have been reported in the literature. Ashrafi [16] suggested that a particular process model, CMM or ISO 9000, contributed to enhance a particular set of product quality characteristics. A similar approach was taken by Pardo et al. [17]. They provided a decision tree to show the influence of CMMI-Dev 1.2 and ISO 90003 on the ISO/IEC 25010 product quality

model. In addition, García-Mireles et al. [18] carried out a harmonization between ISO 25010 and process models, such as ISO/IEC 12207 and CMMI. They found that product quality characteristics are described in activities suggested for elicitation and analysis of requirements.

Finally, García-Mireles et al. [9] found that most of the mappings between process and product models have been performed for security related process models. Other mappings have focused on reliability, maintainability and usability and have been performed on process models such as ISO/IEC 12207 or CMMI. However, the report by García-Mireles et al. does not include process models specifically targeted towards VSEs and the discussion about quality models is a conceptual one. It is then necessary to attain insights into the way software organizations are addressing software product quality. A starting point is identifying the quality characteristics addressed during elicitation and analysis of requirements [18].

2.2. Identifying Relevant Quality Characteristics in Software Projects

Software quality models, such as ISO/IEC 25010, provide a set of quality characteristics that can be used to specify quality requirements, measure software quality and assess product quality [10]. A quality characteristic is a “category of software quality attributes that bear on software quality” [10]. Software quality characteristics can be refined into multiple levels of subcharacteristics and finally into software quality attributes. In daily practice, however, software developers can address quality requirements or nonfunctional requirements. Indeed, a quality requirement is a “requirement that a software quality attribute be present in software” [10]. Thus, there is a close relationship between a quality characteristic and a quality requirement. While the former makes reference to a set of attributes that an entity may have, the latter describes a particular attribute that software must have.

Researchers have studied the perceived significance of different types of quality requirements using product quality models. Models such as ISO/IEC 9126 or ISO/IEC 25010 provide a standardized set of terms and definitions that can be used to: classify quality requirements from a software specification [19], identify relevant quality requirements as regards the type of software developed and different roles in a company [20-22] and assess product quality in particular application domains [23, 24].

One way to assess product quality is to develop a questionnaire based on the quality subcharacteristics definitions [23-26]. In the case of particular domains, the questions are concerned with the relevant concepts of the domain under study [24]. The questionnaires have been rated by both users and experts in either Information Technology or application domain. So, practitioners who work in VSEs can also rate relevant quality requirements based on a questionnaire.

As occurs in large organization, VSEs should address both the functional and the quality requirements in their products, since they determine the software features and behavior [19, 27]. Quality requirements can count for a significant part of project requirements. Svensson et al. [19] found that 38% of the requirements of a requirements specification appertain to the category of quality requirements. Other authors have reported that software project may have up to 20% of quality requirements [20]. Dealing with quality requirements is not only a challenge to large companies but also for small

ones [21]. Therefore, it is expected that VSEs also consider quality requirements in the software they develop.

3. Mapping ISO/IEC 25010 onto ISO/IEC 29110

This Section describes the mapping method carried out, the execution of its activities and the resulting main findings. The purpose is to identify the product quality characteristics addressed in the ISO/IEC 29110 process model and the activities that support them. The information obtained might either provide a basis on which to compare process model features and VSEs' needs as regards product quality or allow the development of a process model adaptation that will lead to the enhancement of the specific product quality characteristics that VSEs require.

3.1. Mapping Method

The method presented herein was based on the method presented in [18]. The main difference is that the method applied in this paper includes the formal identification of the elements under comparison. This change corresponds to one of the practices for preparing standardized profiles [28]. The activities included in the mapping method are:

- Analyze models. The purpose of this activity is to identify the goals of the quality models, describe their structure and requirements.
- Design mapping. The purpose of this activity is to establish a procedure for carrying out the mapping.
- Execute mapping. The purpose of this activity is to perform the mapping between quality models.
- Prepare a report. The purpose is to present a report with the results of the mapping between models.

3.2. Execution of the Mapping

The result of applying the mapping between the process standard (ISO/IEC 29110) and the product quality model described in ISO/IEC 25010 is presented in this Section, which is organized according to the activities described above.

3.2.1 Analyze Models

ISO/IEC 25010 [10] is a standard that describes the quality in use model and the product quality model. The quality in use model has five characteristics, including effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. Its primary concern is quality when software is used in the operation stage of its life cycle. The product quality model identifies eight quality characteristics: functional suitability, performance

efficiency, compatibility, usability, reliability, security, maintainability, and portability. It addresses quality when software is in the development stage.

Fig. 1 depicts the quality characteristics belonging to the product quality model. The number inside the parenthesis shows the number of subcharacteristics included within each of the quality characteristics. For instance, the maintainability quality characteristic includes five subcharacteristics: modularity, reusability, analyzability, modifiability, and testability. In this work, the focus is on the product quality model, since the interest is in addressing product quality during software development.

The ISO/IEC 25010 provides consistent terms and definitions to address relevant quality characteristics of all software products [10]. The quality models can be used to specify, measure, and evaluate software quality. Thus, ISO/IEC 25010 can be applied in different stages of the software lifecycle, included in the requirements, development, use, and maintenance stages.

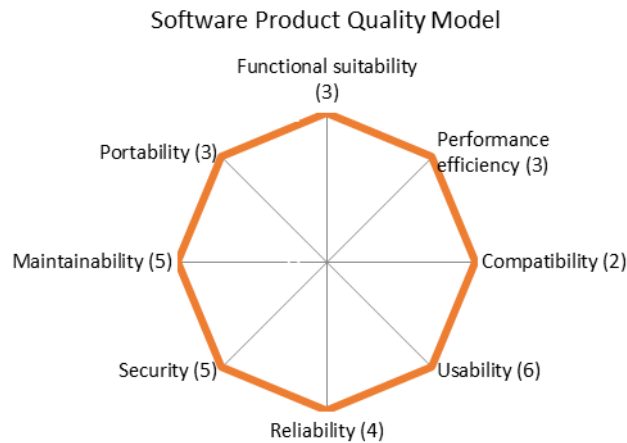


Fig. 1. Quality characteristics from ISO/IEC 25010 [29]

The ISO/IEC 29110 [2] includes a set of standards and technical reports targeted to very small organizations, with the aim of improving both software quality and process performance. The expected benefits include an improved internal management process, enhanced customer satisfaction, improved product quality, and a decrease in development costs. Taking into account the limited resources of very small enterprises, the ISO/IEC 29110 focuses on the project management process and software implementation process. These processes were derived from the ISO/IEC 12207 software lifecycle processes standard [30]. The products described inside these processes rely on the ISO/IEC 15289 information products [11].

A primary component of the ISO/IEC 29110 is the profile that a VSE can apply to implement specific practices, which they do by means of guidelines published as technical reports. A profile is a “set of one or more base standards and/or standardized profiles and, where applicable, the identification of chosen classes, conforming subsets, options and parameters of those base standards, or standardized profiles necessary to accomplish a particular function” [2]. A profile group is a “collection of profiles which

are related either by composition of processes (i.e. activities, tasks) or by capability level, or both” [2]. Each profile must be related to additional components such as an assessment guide and at least one implementation guide. Deployment packages can be developed as optional material to facilitate the implementation of profiles [2].

ISO/IEC 29110 includes the generic profile group that is applicable to VSEs that do not develop critical software products [28]. The profile group is decomposed into initial, basic, intermediate and advanced. Currently, the basic profile is published as a technical report [11]. In this work, the focus is on the Basic Profile ISO/IEC 29110 to determine the best approach for addressing product quality. In order to meet a conformance profile, a VSE must demonstrate that 1) the work product developed is in conformance with content described in the mandatory information products and that 2) the current practices applied in a software project produce the mandatory products described in the profile processes. Process, activities, objectives, work products, and outputs are mandatory. Tasks and inputs are optional.

The basic profile consists of two processes, the project management process and the software implementation process [11]. The purpose of the former process is to establish, and perform systematically, software development activities with the expected quality and within time and cost restrictions. The purpose of the latter process is to systematically carry out the activities of analysis, design, construction, integration and testing of a software product that meets the specified requirements [11]. Both processes are described in terms of purpose, objectives, input products, output products, internal products, roles involved, information diagram flow, activities, and tasks.

3.2.2 Design the Mapping

The mapping purpose is to identify those activities in a process model that address product quality characteristics. Hence, ISO/IEC 25010 was used as a source for product quality vocabulary, terms and definitions [10]. From the ISO/IEC 29110 [11], each process, goal, purpose, process, activity, task and work product was compared with the product quality characteristics. The method used tables to identify each element from the ISO/IEC 29110 standard. Each row contains the description of a process element. If a process element mentioned a product quality characteristic then the row was linked to the appropriate clause from ISO/IEC 25010.

3.2.3 Execute the Mapping

The mapping was executed after the design mapping activity. In order to preserve the relationship between both standards, an Excel worksheet was used for each process element. For instance, the mapping between process objectives from the Software Implementation Process and product quality characteristics is presented in Table 1. The template contains columns that allow the identification of each element, as they are described in the standards, and includes a column to verify if the element description addresses any product quality characteristic. In this case, only the objective SI.O2 mentions two quality sub-characteristics: correctness and testability. The former belongs

to the functional suitability characteristic while the latter belongs to the maintainability quality characteristic. Process objectives from the Project Management Process do not address product quality characteristics; hence, this process is no longer used in this mapping.

The comparison of activities described in ISO/IEC 29110 used the same structure of Table 1. We found that activity SI.2 Software requirements analysis (from ISO/IEC 29110) is related to objective SI.O2 (from ISO/IEC 29110) and this activity also mentions correctness and testability as quality sub-characteristics.

Table 1. Mapping objectives from the Software Implementation Process (from ISO/IEC 29110) with product quality characteristics (from ISO/IEC 25010) [29]

Software Implementation Objective	Brief description	Product Quality?	Product quality characteristic - Reference ISO/IEC 25010
SI.O1	Tasks are performed as regards the project plan	0	
SI.O2	Software requirements are defined and analyzed for correctness and testability , ...	1	4.2.1.2 functional correctness 4.2.7.5 testability
SI.O3	Software architectural and detailed design is developed and baselined	0	
SI.O4	Software components defined by the design are produced	0	
SI.O5	Software components are integrated and verified	0	
SI.O6	Software configuration is integrated and stores at project repository	0	
SI.O7	Verification and validation tasks are performed	0	
Total		1	

When tasks were compared, it was found that two of the seven tasks related to SI.2 software requirements analysis activity (from ISO/IEC 29110) mention quality characteristics (from ISO/IEC 25010). The task SI.2.3 suggests verifying the correctness and testability of the requirements specification. This task is consistent with the objective SI.O2. The task SI.2.4 suggests “to validate that requirements specification satisfies needs and agreed upon expectations, including the user interface usability” [11]. However, usability is not explicitly mentioned in process objectives and activities.

In relation to work products, it was found that requirements specification addresses most of the quality characteristics described into ISO/IEC 25010, except security and compatibility. Software design document suggests addressing performance efficiency, security, usability and reliability. However, there is an inconsistency, since security is not mentioned as content in the requirements document but it is included in the design. The software user documentation and maintenance documentation ask for content understandability. This quality subcharacteristic corresponds to usability.

3.2.4 Prepare a Report

The analysis of mapping between process reference and product quality model showed that ISO/IEC 29110 is aware of the need to improve product quality. From the process perspective, only Objective 2 (SI.O2) of the Software Implementation Process addresses two quality sub-characteristics, named as correctness and testability. These subcharacteristics must be defined and analyzed during the software requirements analysis activity (SI.2), since objectives and activities are mandatory parts of the process, as described in ISO/IEC 29110. The task SI.2.3 suggests verifying and obtaining approval of requirements using a verification approach. In addition, task SI.2.4 recommends validating the usability of the user interface. Although usability is not explicitly addressed in the process objectives, it is important to recognize its relevance for products developed by VSEs.

When focusing on the work products included in the Software Implementation Process, several work products were found that address product quality characteristics: requirements specification, software design, the product operation guide, software user documentation, and the maintenance documentation. Nevertheless, quality characteristics addressed in the requirements specification are very difficult to trace in other work products of the software implementation process. In addition, there is a lack of definition of terms related to quality characteristics.

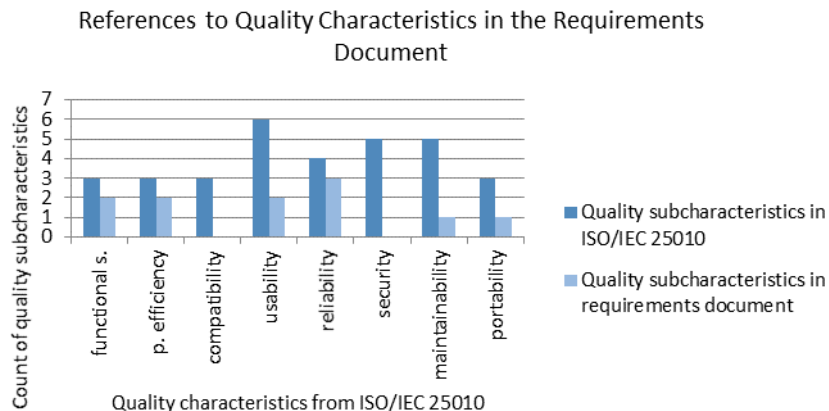


Fig. 2. Extent to which quality subcharacteristics in ISO/IEC 25010 are addressed in the requirements specification document [29]

The requirements specification addresses several quality characteristics and subcharacteristics as they are described in the ISO/IEC 25010 (Fig. 2). These quality characteristics are addressed as quality requirements. It is important to note that the security and compatibility quality characteristics are not addressed in the description of requirements specification document [11]. This inconsistency could provide a basis to harmonize the ISO/IEC 15289 with ISO/IEC 25010.

4. Empirical Study

The purpose of the exploratory survey is to attain insights of the relevant quality characteristics that practitioners of VSEs address in their own software projects. The goal is to contrast the conceptual outcomes of the mapping study, which show that functional correctness and testability must be analyzed in requirements documents, with current industrial practice.

The goal of the study is to identify the relevant product quality characteristics that VSE practitioners consider when developing software. A relevant quality characteristic is mapped onto a customer/user quality need that becomes a quality requirement that development team must fulfill during software development. Customers will, therefore, accept the product when software quality requirements have been implemented.

The study states two research questions: Q1. What product quality characteristics are relevant to VSE practitioners? And Q2. What practices do they carry out to meet these expected quality characteristics?

A survey was conducted since it allows to collect information from people in order to describe, compare or explain their knowledge, attitudes and behavior [31]. It is a retrospective study whose data can be quantitatively or qualitatively gathered by means of interviews or questionnaires [31]. An exploratory empirical study aims to identify new insights with regard to a phenomenon and to generate new hypothesis to be studied [32]. This exploratory survey is a cross sectional where participants are asked for information once [33]. The construction of survey instruments, which are denominated as questionnaires, requires the following steps [34]: search for relevant literature, construct an instrument, evaluate the instrument and document the instrument.

4.1. Developing a Survey Questionnaire

The work reviewed in Section 2 was used as a basis to develop a questionnaire with which to support the exploratory survey. The instrument has three sections. The first part addresses the background information while the second part contains the questions regarding product quality subcharacteristics. Finally, the third part is focused on the instrument evaluation.

The first part requests factors that affect the relevance of quality characteristics. These factors were obtained from [19-22, 35, 36]. In particular, factors considered were the participant's academic degree, the role played in software projects, the type of software developed, and the industrial sector in which the software product is bound to. In addition, the development method, programming languages and specific practices used to achieve the quality requirements were included in the questionnaire. There was also a specific question that asked whether the participant had worked in a VSE in order to determine whether the response would be valid for this study.

The second part of the questionnaire presents questions concerning quality characteristics. The quality characteristics and subcharacteristics were taken from the ISO/IEC 25010 product quality model. Quality subcharacteristics can be used to classify requirements and assess product quality. As Franch et al. [35] pointed out, the "standard quality characteristics and quality subcharacteristics of ISO/IEC 9126 are quite

reasonable and should be used unless a good reason for not doing so comes out during domain analysis". ISO/IEC 25010, as descendant of ISO/IEC 9126, can therefore be used in a similar way in order to discover relevant quality characteristics.

The product quality model is composed of eight quality characteristics which have a total of 31 subcharacteristics. For each quality subcharacteristic a yes/no question was formulated considering the way other instruments worded quality-related questions [23-25]. Appendix A includes the questions used to identify relevant quality subcharacteristics.

The third part of the questionnaire includes four questions aimed to the utility of it in eliciting information about product quality characteristics and to improve it. It includes questions about the suitability of the questionnaire to identify relevant quality characteristics addressed in software projects (Q1), the extent to which the questions are easy to understand (Q2), the extent to which the questions can be easily related to software project quality requirement (Q3), and the importance of improving product quality in software projects (Q4). These four questions are answered in a five-option Likert scale: strongly disagree, disagree, neutral, agree, and strongly agree. These labels correspond to number 1 to 5, in the same order. There is also an open question that asks for suggestions on how to improve the questionnaire.

4.2. Evaluating the Questionnaire in Interviews

Three semi-structured interviews were carried out with doctorate students who have worked as lecturers on Computer Science Bachelors' degree programs at Mexican universities. Two of the interviewees had research topics related to software quality and all of them had worked at a VSE before.

It took roughly half an hour to conduct the interviews. As a result of using the questionnaire, interviewees identified relevant quality characteristics that they had to address when they worked in a VSE. With regard to the assessment of the questionnaire, all interviewees agreed that the instrument was suitable for identifying relevant quality characteristics for VSEs, the questions were easy to understand and the instrument was relevant as regards improving product quality in the context of VSEs. They also suggested that the questionnaire could be improved by: conducting face-to-face interviews in order to clarify doubts concerning questions, translating the questionnaire into Spanish, adapting the model to contextual settings and providing contextual data for each question.

Two of these suggestions were followed, so the questionnaire was translated into Spanish, since the population in the survey consisted of Mexican software developers, and the background information was enriched with contextual data and examples to make the questions clear. However, the model was not adapted to contextual settings since we were targeting any VSE established in Mexico which may, therefore, develop software for very different domains.

4.3. Collecting Data

Existing academic-industry relationships were used as a basis to select a set of participants in an opportunistic manner. The main requirement for participation was that they had to be composed of up to 25 people and not to develop critical software. Collecting data procedure was carried out in two stages.

In the first stage, the exploratory survey was supervised, i.e., the researcher ensures that respondents understand each question and provide an answer [33]. Seven interviewees, who worked in two different companies were surveyed in July 2016. The analysis of the responses provided in the questionnaire's third Section showed that interviewees agreed on the utility of the questionnaire to identify relevant quality characteristics in software projects and the questions were easy to understand. In addition, these seven participants reported that questions could be easily related to quality requirements.

As a result of the first stage, the data collection in the second stage was conducted in an unsupervised way. The survey was sent via email to former Computer Science major students from Mexican universities that were invited to participate. 46 of them completed the survey, adding up a total of 53 responses.

4.4. Response Profile (Organization, Project, Participant)

In this study, 53 surveys were received. Three surveys were dropped since they did not meet the quality criteria. One of them was filled in by a software architect who is part of a team of 80 people. The second paper was discarded because the participant did not develop software. Finally, the third survey was discarded because the file was corrupted. Thus, 50 valid surveys were obtained.

The data was categorized as regards the organization reported by participants. Those that provided the company name were labeled with a letter to not disclose sensitive information. The name of the company was not a required answer in the questionnaire so, 13 respondents did not provide this information (see Fig. 3 top row, N/A – Not available). In addition, two participants reported that they worked as freelance workers. 13 distinct organizations were sampled by only one participant and five organizations were sampled from two to seven participants.

The participants provided information about the entity size where they worked. Based on the definition of VSE, the questionnaire included a yes/no question about if s/he worked in a VSE. The questionnaire also included the entity categories: enterprise, department or project. On the one hand, 14 responses were labeled as No-VSE since they had more than 25 employees (see Table 2). At project level, however, the project teams were composed from 1 to 15 people, team size consistent with the definition of a VSE [2]. On the other hand, 36 responses were labeled as VSE. In this category, both the entity and project team sizes were consistent with the definition of a VSE.

The participants provided quality certification information about their companies. Half of responses pointed out that the company has no certification. 27% responses reported one certification while roughly 15% (7 out of 50) reported that company had

three different certifications. The most common mentioned models or standards were ISO 9000, CMMI, and MoProSoft.

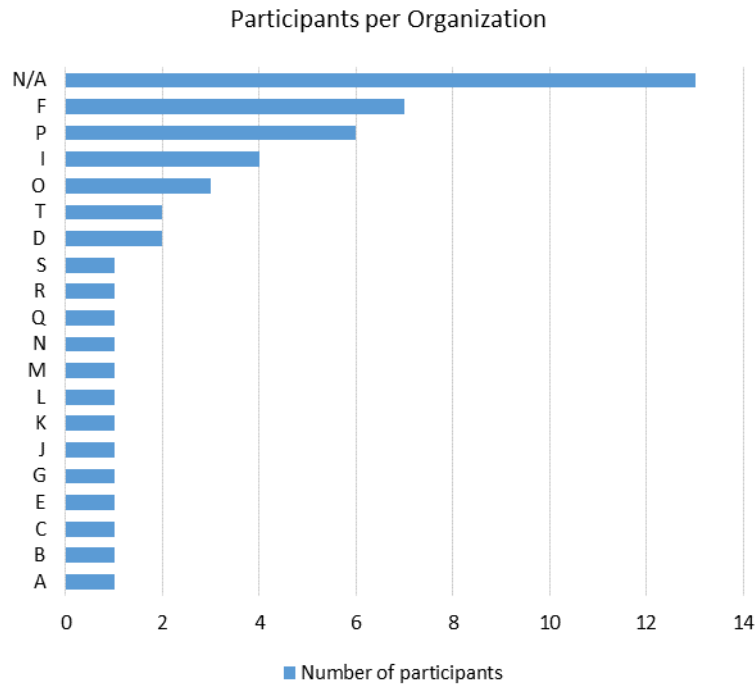


Fig. 3. Frequency of participants per organization

Table 2. Entity size

Category	Number of responses	Scope	Frequency scope	Entity size	Project team's size
No VSE	14	Enterprise	12	30 up to 5000	1 up to 15
		Department	2	14 up to 60	3
VSE	36	Enterprise	12	4 up to 25	2 up to 20
		Department	8	4 up to 20	1 up to 3
		Project	6	2 up to 9	1 up to 6
		Not labeled	9	1 up to 15	1 up to 2

Most of the participants in this survey (66%) had an engineer or bachelor’s degree in Computer Science. Four participants have a PhD degree in Computer Science and only one had it in Nanotechnology while eight participants had a Mater in Science in

Computer Science and two more had it in Mathematics. Finally, two participants had completed only two years of their computer science bachelor degree.

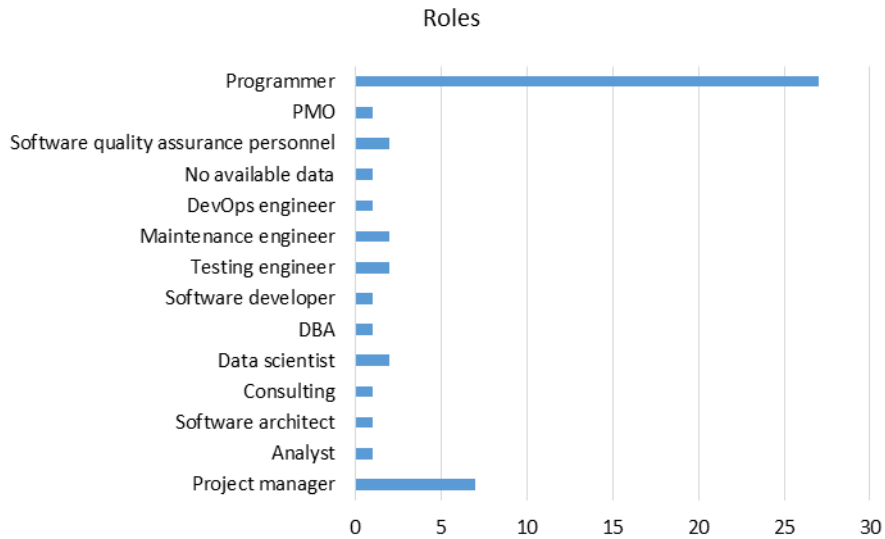


Fig. 4. Participants' role

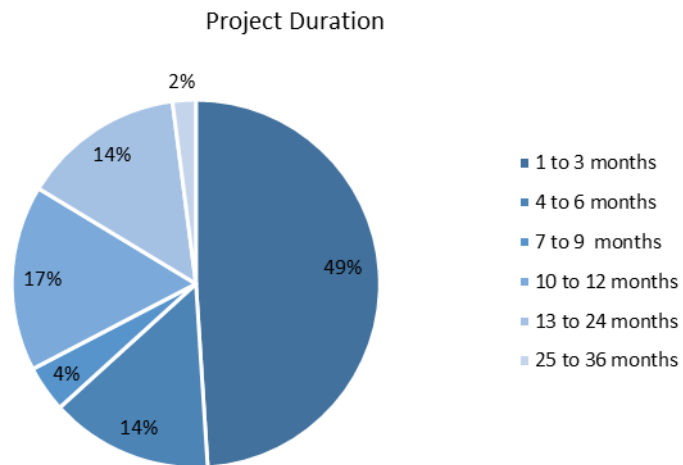


Fig. 5. Project duration in months

Participants had different responsibilities, as shown in Fig. 4. In this question, participants only had to choose (or write) one option. They reported 13 distinct roles. One participant did not provide data in this question. More than half of the participants

worked as programmers (27 out of 50 participants) and only seven participants had responsibilities as project managers. Other new roles appeared such as DevOps engineer and Data scientist, which would require special practices to address software product quality. As regards the experience working in the role reported, the range is between 5 to 192 months with an average of 46 months, almost four years.

The type of software participants developed was as follows: 50% corresponds to web applications while 6% of the participants write they developed mobile applications. 26% of the responses were focused on desktop applications while 6% of the respondents worked with embedded systems. In addition, 8% worked with applications that needed to be available in several channels, such as web, mobile or desktop. The main industrial sectors where software projects were developed correspond to commercial services (28%), education (24%) and health (14%).

Project addressed by participants lasted from 1 to 36 months. Fig. 5 depicts that half of the respondents worked in projects that spanned between 1 and 3 months. Around 20% of the participants worked in projects that span from 7 to 12 months. 14% of the participants worked in projects which spans for two years while 2% of participant worked in three-year projects. The median of project duration was 4 months.

4.5. Relevant Quality Characteristics for Surveyed Participants

Almost all participants answered the second part of the questionnaire on relevant quality characteristics. From 50 valid responses, seven questionnaires had an unanswered one question while an additional questionnaire had two questions without answers. This represents only 0.58% of incomplete data. Fig. 6 shows no responses in learnability, operability, user error protection, maturity, fault-tolerance, non-repudiation, accountability, authenticity and modifiability.

Table 3. Most common quality subcharacteristics

Quality characteristic	Quality Subcharacteristic	Percentage
Functional Suitability	Functional Completeness	90.00
	Functional Correctness	92.00
	Functional Appropriateness	98.00
Performance Efficiency	Time Behavior	88.00
	Resource Utilization	88.00
Reliability	Availability	96.00

As regards the frequency of quality characteristics addressed in software projects, Fig. 6 depicts answers based on quality subcharacteristics. The figure shows the frequency of yes, no, and no data (response) categories for each quality subcharacteristic. The most common subcharacteristic considered by participants was functional appropriateness (49 out of 50 responses) which belongs to functional suitability quality characteristic.

The less common subcharacteristic addressed in software projects was accessibility (9 out of 50 responses), according to participants' perceptions. From the 31 quality subcharacteristics, one participant only considered 5 (out of 31) of them relevant to her/his software project. On the other hand, another participant considered 29 (out of 31) relevant quality subcharacteristics.

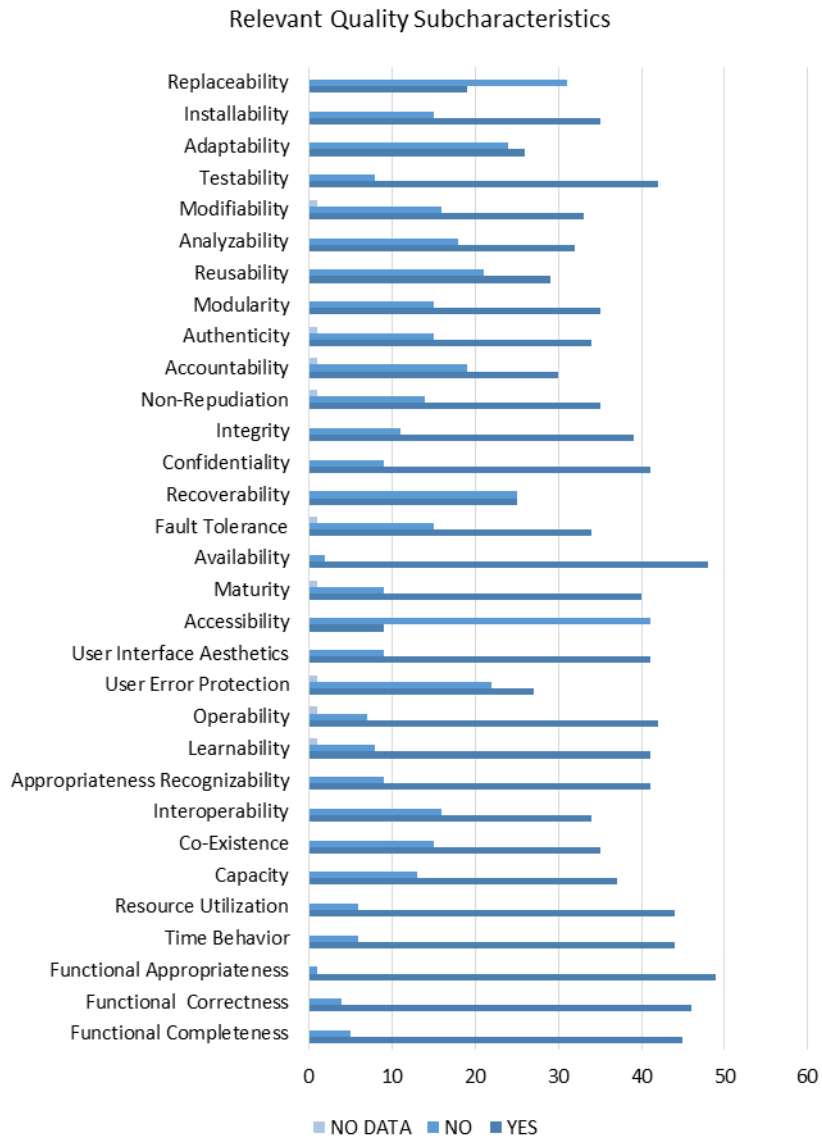


Fig. 6. Quality subcharacteristics addressed in projects performed by surveyed participants

In order to identify the most relevant quality characteristics, an 86% upper limit was used arbitrarily to select the most relevant quality characteristics. This value is consistent when practices are assessed as regards process capabilities where more than 85% represents a fully established practice. This approach have been used to identify the level of harmonization between process models [37]. Table 3 shows that functional suitability, performance efficiency and reliability are quality characteristics relevant for surveyed participants. The results showed that all quality subcharacteristics belonging to functional suitability were important in the software projects addressed by participants. Performance efficiency has only two (out of three) relevant subcharacteristics while reliability only has one (out of four). Operability (subcharacteristic of usability) and testability (subcharacteristic of maintainability) achieved 84% of relevance in software projects.

The less common quality subcharacteristics were accessibility that was relevant only to 18% of the participants and replaceability that was important to 38% of the participants. Since these participants were not working with critical systems and projects time span were short, there was a reasonable the lack of attention to reliability, maintainability and portability.

4.6. Practices Used To Address Product Quality

In order to identify used practices to improve software product quality, the questionnaire had questions about software development methods and practices. In the former (Fig. 7), results depicted a variety of approaches to develop or maintain software. Agile methods, including the adapted ones, represents 42% of the survey's responses. Incremental methods without reference to agility (Unified Process, Incremental, Spiral) counts nearly 18 of responses. Around 18% used a method adapted to specific project or organizational needs. The cascade method was used by 14% of participants. In addition, 10% of the responses pointed out that in some projects the main software development approach is deploying software construction activities.

Participants reported several practices deployed in software projects. The analysis of responses focused on those activities directly related with software product quality, mainly those described in Bourque and Fairley [38] related to the following areas: software requirements, software quality, and software testing. 98% of the participants reported that they deployed between one up to four practices in their projects. From this responses set, 34% deploys three practices.

The three most common practices were related to requirements, testing and customer validation (see Fig. 8). Requirements related practices (24% of the respondents) is a category that includes both formal and informal gathering of requirements and any method used to analyze them. If participants mentioned in their responses terms like 'nonfunctional requirement' or 'quality requirements' or any quality attribute, these were categorized as quality requirements focused (6%). As regards testing practices, 21% of the participants pointed out that they carried out testing activities based on software requirements. Some participants highlighted that the testing approach sometimes is an informal one. The category labeled quality specific testing (Fig. 8) depicts 9% of the participants executed testing procedures for specific product quality characteristics such as performance, stress, security and usability. The third most

important approach to address product quality is based on client validation. Customer feedback was used by 12% of the participants to validate they met the requirements, including the quality ones. Other practices mentioned were informal reviews, code reviews, and design reviews.

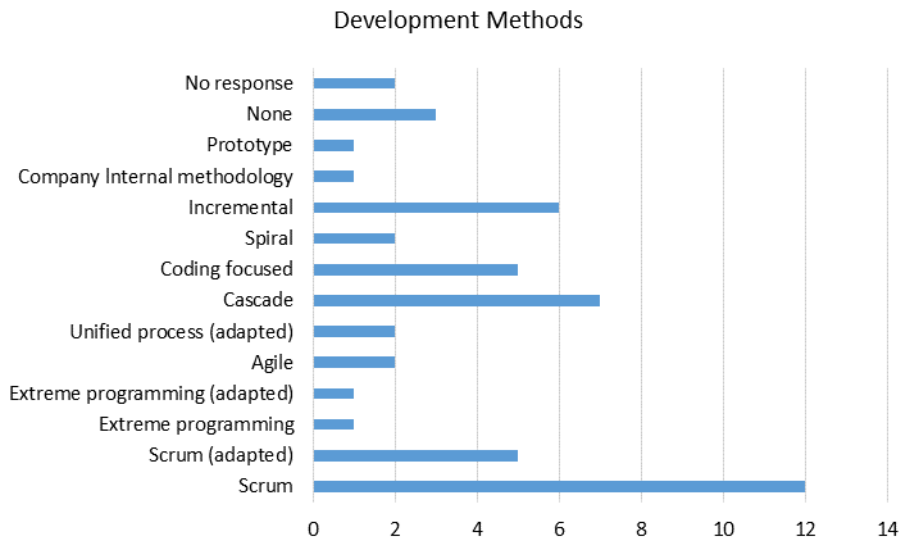


Fig. 7. Development methods reported by participants

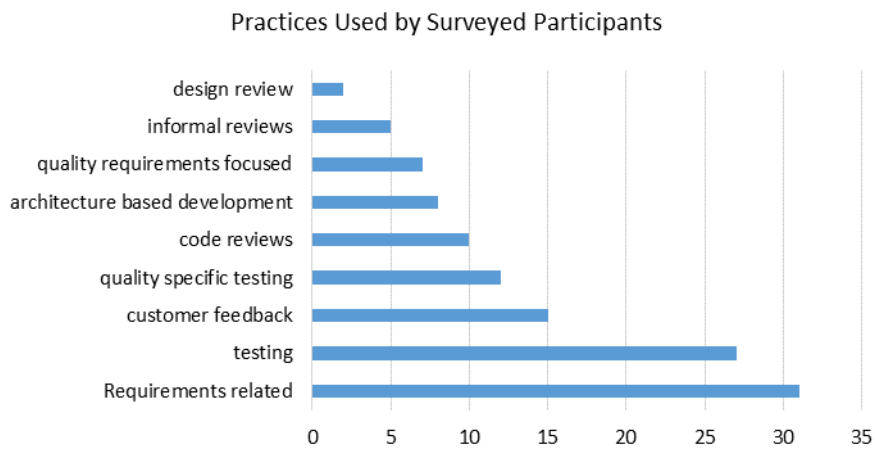


Fig. 8. Practices deployed in software projects to enhance product quality

4.7. Questionnaire Perceptions

Part three of the questionnaire is focused on the participants' perception about its quality. Fig. 9 illustrates the results ordered by the selected responses from the five-point Likert scale. Q1 asked for the relevance of the questionnaire to identify quality characteristics in software projects. As a result, the median was four, corresponding to 'agree'. Q2 addressed the ease to understand the questions and the results showed a median of 4 ('agree'). Q3 asked for the extent to which quality characteristics' related questions were easy to relate with software quality requirements. Again, the median was 4 (agree) for Q3. In Q4, median was 5, which means that participant are strongly agree that enhancing product quality is important in their software projects. On the other hand, strongly disagree (1) and disagree (2) categories were selected by very few participants. Neutral (3) category was chosen by almost 20% of the participants from questions one to three. Most of the responses were in the agree (4) and strongly agree (5) categories. Thus, these values may suggest that the questionnaire was appropriate to study quality characteristics in the context of VSEs.

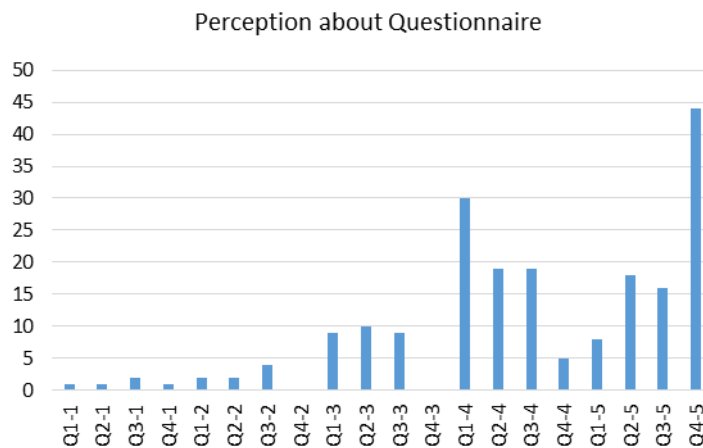


Fig. 9. Perception on questionnaire quality

The last question in the questionnaire asked for suggestions to improve the instrument. As a summary of 13 comments, participants mentioned that the survey should be online, the questions should be more specific as regards the type of software projects, and the number of questions should be reduced. Others suggested to include new trends in software development and strategies used to improve product quality. For some participants, the open question about practices used to enhance product quality was difficult to answer. Although the questionnaire's responses showed its appropriateness for this study, there are also improvement opportunities.

4.8. Threats to Validity

The exploratory survey provided information on the relevant quality characteristics practitioners consider when they develop software in VSEs. Although the guidelines required to conduct surveys were followed [33], there are several limitations in this type of empirical research. In relation to external validity, this study was exploratory and the participants were selected by a contact list from two Mexican universities. The list of relevant quality characteristics cannot, therefore, be generalized to other organizations. However, the results provide an initial point to formulate appropriate hypothesis in the context of product quality and VSEs. In addition, the results provide initial insights about the most relevant quality characteristics for participants. For instance, Table 3 shows the most common quality subcharacteristics addressed by participants.

Several contextual factors need to be considered in order to determine the most important quality characteristics [20, 22, 36]. The first part of the questionnaire provided data about the entity size and asked for any quality certification, the academic degree of participants, the role played by participants in their last projects and their experience in that role. Most of the participants carried out software construction activities and few worked in other software engineering disciplines. Thus, selecting a random sample is needed to validate the distribution of roles in the population. In addition, the identification of functionality and performance efficiency as the most relevant quality characteristics is consistent with other studies [19, 39]. Furthermore, other researchers have found that functional suitability, performance and reliability are the most important quality attributes while portability and installability are the least relevant [40].

In this study, reliability was improved by using a standard product quality model as a basis to identify relevant quality characteristics. Both the terms and the definitions in ISO/IEC 25010 are accepted by the software engineering community. However, some researchers have recommended adapting the quality model to the application domain under study [19, 21]. In this work, it was not possible to adapt the quality model since VSEs can develop software in any application domain. In order to verify clear understanding of the questions, an evaluation was, therefore, conducted with experts in the development of software and product quality. They assessed and validated the instrument. In addition, the items in the background information section were derived from current literature concerning factors that affect software quality [20, 36].

Construct validity was, meanwhile, improved by adding contextual information to the questions and rewriting some of them in accordance with the findings of the survey evaluation. During the interviews, some questions were clarified in order to facilitate the participants' responses. However, this study was focused on the practitioners' perceptions of the relevant product quality characteristics. It is thereby necessary to investigate objective evidence concerning the type of quality requirements that practitioners address. For instance, documents created in software projects can be analyzed as regards the quality requirements included [19].

5. Discussion

The implementation of process models focused on small companies' profiles can improve software quality. This goal can be achieved by implementing the set of practices suggested in the ISO/IEC 29110. In addition, this standard shows that small companies also need to be aware of product quality characteristics. It is necessary to define and verify the correctness of the software specification and testability characteristics. However, the standard does not provide a definition of each product quality characteristic. Hence, ISO/IEC 25010 can be used to establish a baseline vocabulary for managing software product quality.

From the two processes described in ISO/IEC 29110, only the software implementation process refers explicitly to product quality characteristics. However, the mandatory process elements only address quality requirements during the validation activities of the requirements document. When the requirements document is analyzed, other quality characteristics are mentioned as optional. In addition, the design document, the maintenance document, and the user documentation address several optional quality characteristics.

In relation to the empirical study, the results show that practitioners at VSEs are interested in product quality. The main quality characteristics are functional suitability, performance efficiency, and reliability. This is consistent with reports in other studies that indicate the same quality characteristics as being the most relevant [19, 39]. However, several factors affect software quality [36] and they must be considered before an improvement process program is deployed.

Practitioners use general software engineering practices to address the relevant quality characteristics. Indeed, this simplified means of developing software is consistent with the characteristics of small organizations [3, 5]. Most of the participants deployed agile or incremental methods in their projects. Main practices are related to gathering and documenting requirements, testing software to validate requirements and customer feedback. Few participants reported specific practices focused on particular quality characteristic. As Phillips et al. [21] noted: "quality and functionality are tightly linked and therefore that majority of the organizations did not have a separate process for the management of [quality requirements]".

A process improvement initiative based on the ISO/IE 29110 Basic Profile is a first step towards addressing product quality, and particularly towards enhancing the functional correctness and testability of software requirements. Indeed, the practitioners also consider these quality subcharacteristics relevant (testability was important for 84% of the participants). As the empirical study shows, VSE practitioners are aware of several more product quality characteristics and they are committed to develop a high quality software product. However, they need practices to ensure that the quality characteristics are properly addressed during the software development life cycle. This insight may, accordingly, foster further research into the way product quality should be managed in VSEs and the development of new profiles that contribute to enhance product quality.

6. Conclusions

Very small organizations are interested in enhancing product quality. They can implement improvement programs on the basis of both quality views: process and product. In this work, product and process models were compared to identify the extent to which both quality views are taken into account. In order to attain insights into the relevance of product quality for VSEs, an exploratory survey was conducted, which showed that VSEs also need support to enhance product quality.

The comparison between models was carried out by means of a mapping. The results of the mapping show that ISO/IEC 29110 considers the activity employed to analyze correctness and testability of software requirements documents to be mandatory. As recommended in the ISO/IEC 29110, usability is considered in the context of analyzing user interfaces. However, the documents suggest several of the quality characteristics in ISO/IEC 25010, but they do not include security or compatibility.

The empirical study shows that VSEs need support to address product quality. Practitioners are committed to product quality and they consider several quality characteristics (mainly functional suitability, performance efficiency, and reliability) to be relevant. However, they need practices with which to elicit, specify, analyze, and validate quality requirements that will take into account their limited resources.

Given the practitioners' interest in product quality and the results of the mapping, there is a need for further research to provide VSEs with process models that support both the introduction of effective practices to improve the process and the provision of specific practices to enhance specific product quality characteristics. Developing those guidelines may allow VSEs to refine their product quality strategy. Furthermore, these models and guidelines should be validated with empirical studies.

References

1. Larrucea, X., Connor, R.V.O., Colomo-Palacios, R., Laporte, C.Y.: Software Process Improvement in Very Small Organizations. *IEEE Software*. 33, 85-89 (2016)
2. ISO: ISO/IEC 29110-1:2011 Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 1: Overview. (2011)
3. Basri, S., O'Connor, R.V.: Understanding the perception of very small software companies towards the adoption of process standards. In: Riel, A., O'Connor, R.V., Tichkiewitch, S., Messnarz, R., (eds.). *Communications in Computer and Information Science, CCIS*. 99, pp. 153-164. Springer Berlin Heidelberg: Grenoble, France. (2010)
4. Richardson, I., von Wangenheim, C.G.: Why are small software organizations different? *IEEE Software*. 24, 18-22 (2007)
5. Habra, N., Alexandre, S., Desharnais, J.-M., Laporte, C.Y., Renault, A.: Initiating software process improvement in very small enterprises: Experience with a light assessment tool. *Information and Software Technology*. 50, 763-771 (2008)
6. Muñoz, M., Gasca, G., Valtierra, C.: Caracterizando las necesidades de las pymes para implementar mejoras de procesos software: una comparativa entre la teoría y la realidad. *Revista Ibérica de Sistemas y Tecnologías de la Información*. E1, 1-15 (2014)
7. Laporte, C.Y., O'Connor, R.V.: A systems process lifecycle standard for very small entities: Development and pilot trials. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R.,

- (eds.). *Communications in Computer and Information Science*, CCIS. 425, pp. 13-24. Springer Berlin Heidelberg: Luxembourg. (2014)
8. Oktaba, H., García, F., Piattini, M., Ruiz, F., Pino, F.J., Alquicira, C.: Software process improvement: The competisoft project. *Computer*. 40, 21-28 (2007)
 9. García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M.: Approaches to promote product quality within software process improvement initiatives: A mapping study. *Journal of Systems and Software*. 103, 150-166 (2015)
 10. ISO: ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQaURE) - System and software quality models. (2010)
 11. ISO: ISO/IEC TR 29110-5-1-2:2011 Software engineering — Lifecycle profiles for Very Small Entities (VSEs) —Part 5-1-2: Management and engineering guide:Generic profile group: Basic profile (2011)
 12. Kelemen, Z.D., Kusters, R., Trienekens, J.: Identifying criteria for multimodel software process improvement solutions-based on a review of current problems and initiatives. *Journal of software: Evolution and Process*. 24, 895-909 (2012)
 13. Pardo, C., Pino, F.J., García, F., Baldassarre, M.T., Piattini, M.: From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies. *Journal of Systems and Software*. 86, 125-143 (2013)
 14. Henderson-Sellers, B.: Standards harmonization: Theory and practice. *Software and Systems Modeling*. 11, 153-161 (2012)
 15. Pardo, C., Pino, F.J., García, F., Velthuis, M.P., Baldassarre, M.T.: Trends in harmonization of multiple reference models. In: 5th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 61-73. Springer-Verlag Berlin, Athens, Greece (2011)
 16. Ashrafi, N.: The impact of software process improvement on quality: In theory and practice. *Information and Management*. 40, 677-690 (2003)
 17. Pardo, C., Pino, F.J., García, F., Piattini, M.: Harmonizing quality assurance processes and product characteristics. *Computer*. 44, 94-96 (2011)
 18. García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M.: Towards the harmonization of process and product oriented software quality approaches. In: Winkler, D., O'Connor, R., Messnarz, R., (eds.). EuroSPI2012, CCIS 301, CCIS. 301, pp. 133-144. Springer Berlin Heidelberg: Vienna. (2012)
 19. Berntsson Svensson, R., Olsson, T., Regnell, B.: An investigation of how quality requirements are specified in industrial practice. *Information and Software Technology*. 55, 1224-1236 (2013)
 20. Berntsson Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R.: Quality requirements in industrial practice-an extended interview study at eleven companies. *IEEE Transactions on Software Engineering*. 38, 923-935 (2012)
 21. Phillips, L.B., Aurum, A., Svensson, R.B., Managing software quality requirements, in 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA) / 15th EUROMICRO Digital Systems Design (DSD) Cortellessa, V., Muccini, H., Demirors, O., Editors. (2012), IEEE: Cesme, Turkey. p. 349-356.
 22. de la Vara, J.L., Wnuk, K., Berntsson-Svensson, R., Sánchez, J., Regnell, B.: An Empirical Study on the Importance of Quality Requirements in Industry. In: SEKE, pp. 438-443. (2011)
 23. Chua, B.B., Dyson, L.E.: Applying the ISO 9126 model to the evaluation of an e-learning system. In: Proc. of ASCILITE, pp. 5-8. (2004)
 24. Haslinda, N.N., Fariha, M.A.Z., Fahmy, A.W.S., Roslina, W.O.W., Sukinah, A.A.N., Suhana, S.N., Shiratuddin, N.: Refinement of the ISO 9126 model for evaluating software product quality in e-Book. *Australian Journal of Basic and Applied Sciences*. 8, 29-34 (2014)

25. Horgan, G., Khaddaj, S.: Use of an adaptable quality model approach in a production support environment. *Journal of Systems and Software*. 82, 730-738 (2009)
26. Aydin, A.O.: A New Way to Determine External Quality of ERP Software. In: Murgante, B., et al., (eds.). *Computational Science and Its Applications – ICCSA 2012: 12th International Conference, Salvador de Bahia, Brazil, June 18-21, 2012, Proceedings, Part IV*. 7336, pp. 186-201. Springer Berlin Heidelberg: Berlin, Heidelberg. (2012)
27. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J., *Non-functional requirements in software engineering*. 2000: Kluwer Academic Publisher.
28. ISO: ISO/IEC FDIS 29110-2 Software engineering — Lifecycle profiles for Very Small Entities (VSEs) —Part 2: Framework and taxonomy. (2010)
29. García-Mireles, G.A.: Addressing Product Quality Characteristics Using the ISO/IEC 29110. In: Mejia, J., Munoz, M., Rocha, Á., Calvo-Manzano, J., (eds.). *Trends and Applications in Software Engineering: Proceedings of the 4th International Conference on Software Process Improvement CIMPS'2015*. pp. 25-34. Springer International Publishing: Cham. (2016)
30. ISO: ISO/IEC 12207 Systems and software engineering — Software life cycle processes. (2008)
31. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslén, A., *Experimentation in Software Engineering*. 2012: Springer Heidelberg.
32. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*. 14, 131-164 (2009)
33. Kitchenham, B., Pfleeger, S.L.: Principles of survey research part 2: designing a survey. *SIGSOFT Softw. Eng. Notes*. 27, 18-20 (2002)
34. Kitchenham, B., Pfleeger, S.L.: Principles of survey research: part 3: constructing a survey instrument. *SIGSOFT Softw. Eng. Notes*. 27, 20-24 (2002)
35. Franch, X., Carvallo, J.P.: Using quality models in software package selection. *IEEE software*. 20, 34-41 (2003)
36. Curcio, K., Malucelli, A., Reinehr, S., Paludo, M.A.: An analysis of the factors determining software product quality: A comparative study. *Computer Standards & Interfaces*. 48, 10-18 (2016)
37. Pino, F.J., Baldassarre, M.T., Piattini, M., Visaggio, G.: Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504. *Journal of Software Maintenance and Evolution*. 22, 279-296 (2010)
38. Bourque, P., Fairley, R., *Guide to the Software Engineering Body of Knowledge (SWEBOK Version 3)*. 2014: IEEE Computer Society.
39. Sibisi, M., Van Waveren, C.C., A process framework for customising software quality models, in 8th IEEE Africon Conference. (2007), IEEE: Windhoek, NAMIBIA. p. 1-8.
40. Wagner, S., Lochmann, K., Winter, S., Goeb, A., Klaes, M., Nunnenmacher, S. *Software Quality Models in Practice: Survey Results*. 2012 Oct, 2013]; Available from: <http://mediatum.ub.tum.de/doc/1110601/1110601.pdf>.

Gabriel Alberto García-Mireles holds a PhD in Computer Science by the University of Castilla – La Mancha since 2014. He is a lecturer at the University of Sonora in Mexico. He is interested in software product quality, software process improvement and sustainable development.

Received: August 9, 2016; Accepted: November 8, 2016.

Appendix A. Questionnaire used to identify relevant quality characteristics

In this part we wish to discover what quality aspects are relevant for the software under development.

In software projects is relevant for your users and customers that:

Number	Question	Response	
1	The set of functions implemented in software covers all specified tasks and user objectives	Yes	No
2	The software provides the expected results with the appropriate precision	Yes	No
3	The set of functions implemented in software facilitates the accomplishment of specified user tasks and goals	Yes	No
4	When software is in operation, it meets requirements of response time, processing time and throughput rates	Yes	No
5	When software is in operation, it meets requirements regarding the use of system resources	Yes	No
6	When software is in operation, it meets requirements for maximum limits	Yes	No
7	When software is in operation, it efficiently shares common resources and environments with other systems	Yes	No
8	Software must exchange information and use it	Yes	No
9	Users can recognize whether a piece of software is appropriate for their needs	Yes	No
10	Users can easily achieve specified goals of learning to use the product	Yes	No
11	Software is easy to operate	Yes	No
12	Software protects user against making errors	Yes	No
13	User interface enables a satisfying user interaction	Yes	No
14	Software provides functions to be used by people with the widest range of characteristics and capabilities (e.g., children, older adults, people suffering from blindness)	Yes	No
15	Software executes successfully under normal operation	Yes	No
16	Software is accessible when it is needed	Yes	No
17	When faults arise, software operates as intended	Yes	No
18	When faults arise, software can recover data and establish the desired system state	Yes	No
19	Software must ensure that data are accessible only to those authorized to have access	Yes	No

20	Software prevents unauthorized access to programs and data and also prevents their modification	Yes	No
21	Software must prove that actions or events take place	Yes	No
22	Software must ensure actions of an entity are only traced to that entity	Yes	No
23	Software must ensure the identity of a subject or resource is that which it claims to be	Yes	No
24	Software component has minimal impact on other components when it changes	Yes	No
25	Software artifacts, such as components, plans, designs and test cases, can be used in more than one system	Yes	No
26	Assessing the impact of change or identifying the components to be modified is easy	Yes	No
27	Software needs to be modified without introducing defects	Yes	No
28	Test criteria must be established when software is under development/maintenance	Yes	No
29	Software must be easily adapted to different hardware or software or usage environment	Yes	No
30	Software must be easily installed/uninstalled in a specified environment	Yes	No
31	Software must be replaced with another specified piece of software for the same purpose in the same environment	Yes	No
