UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Revista Científica

http://revistas.udistrital.edu.co/ojs/index.php/revcie/index

Artículo de revisión

Identifying Software Quality Attributes Relevant for Very Small Entities

Identificación de atributos de calidad de software para entidades muy pequeñas

Identificação de atributos de qualidade de software para entidades muito pequenas

Mirna Muñoz[1]
Gabriel Alberto García-Mireles[2]
Jezreel Mejía[3]
Elizabeth Villanueva[4]

Recibido: 19 de diciembre de 2025 **Aceptado:** 23 de febrero de 2026

Para citar este artículo: Muñoz, M., García-Mireles, G. A., Mejía, J. M. y Villanueva, E. (2026). Identifying Software Quality Attributes Relevant for Very Small Entities. *Revista Científica*, *53*(1), 1-19. https://doi.org/10.14483/23448350.24627

Abstract

Nowadays, software systems are increasingly important and are rapidly growing in complexity and size. Their development involves large, medium, small, and very small entities (VSEs). The latter play a critical role in the supply chain, so developing high-quality software in accordance with established standards is fundamental to ensure the success of their software products. Applying standards such as ISO/IEC 29110 supports VSEs by providing efficient processes and increased productivity. However, standards such as ISO/IEC 25010, which provide a quality model for software products and quality in use, are not intended for implementation in VSEs. This paper presents an exploratory study that identifies the relevant quality characteristics and sub-characteristics in VSEs, as well as the practices they use to achieve product quality.

Keywords: software process quality, software product quality, ISO/IEC 29110, ISO/IEC 25010, ISO/IEC 25040

1. Centro de Investigación en Matemáticas A.C., Zacatecas, México. Email: mirna.munoz@cimat.mx
2. Department of Mathematics, Universidad de Sonora, Sonora, México. Email: gabriel.garcia@unison.mx
3. Centro de Investigación en Matemáticas A.C., Zacatecas, México. Email: jmejia@cimat.mx
4. Centro de Investigación en Matemáticas A.C., Zacatecas, México. Email: elizabeth.villanueva@cimat.mx

Resumen

En la actualidad, los sistemas de software son cada vez más importantes hoy en día y su complejidad y tamaño aumentan rápidamente. Su desarrollo involucra a entidades grandes, medianas, pequeñas y muy pequeñas. Estas últimas desempeñan un papel fundamental en la cadena de suministro, por lo que desarrollar software de alta calidad conforme a los estándares establecidos resulta clave para garantizar el éxito de sus productos. La aplicación de estándares como ISO/IEC 29110 apoya a las Entidades Muy Pequeñas (EMP) al proporcionar procesos más eficientes y mayor productividad. Sin embargo, estándares como ISO/IEC 25010, que proporcionan un modelo de calidad para productos de software y para la calidad en uso, no están destinados a su implementación en EMP. Este artículo presenta un estudio exploratorio que identifica las características y subcaracterísticas de calidad relevantes en las EMP, así como las prácticas que emplean para lograr la calidad del producto.

Palabras clave: calidad del proceso de software, calidad del producto de software, ISO/IEC 29110, ISO/IEC 25010, ISO/IEC 25040.

Resumo

Atualmente, os sistemas de software são cada vez mais importantes e estão crescendo rapidamente em complexidade e tamanho. Seu desenvolvimento envolve grandes, médias, pequenas e microempresas (EMPs). Estas últimas desempenham um papel crucial na cadeia de suprimentos, portanto, o desenvolvimento de software de alta qualidade, em conformidade com padrões estabelecidos, é fundamental para garantir o sucesso de seus produtos. A aplicação de normas, como a ISO/IEC 29110, auxilia as EMPs, proporcionando processos mais eficientes e maior produtividade. No entanto, normas como a ISO/IEC 25010, que fornecem um modelo de qualidade para produtos de software e qualidade em uso, não são destinadas à implementação em EMPs. Este artigo apresenta um estudo exploratório que identifica as características e subcaracterísticas de qualidade relevantes em EMPs, bem como as práticas que utilizam para alcançar a qualidade do produto.

Palavras-chaves: Qualidade do processo de software, qualidade do produto de software, ISO/IEC 29110, ISO/IEC 25010, ISO/IEC 25040.

INTRODUCTION

Enhancing competitiveness through international models and standards in software development organizations has been a global priority, given their importance in meeting the needs of the software market, and Mexico is no exception in this regard. To achieve this goal, numerous models and standards have been developed, including ISO 9000-3, ISO 12207, ISO 15504, CMM, CMMI, IDEAL, SCAMPI, and CompetiSoft (Mellado Fernandez *et al*, 2015). These models and standards were developed for large companies. However, around 94-97% of companies are small organizations and very small entities (VSEs).

These types of companies perceive models and standards as very heavy and expensive (Sanchez-Gordon *et al.*, 2015). VSEs lack opportunities, including access to trained personnel and necessary resources, for productivity and survival (López, 2018). In addition to not being recognized for producing a quality system or its elements, this may represent an obstacle to their growth (International Organization for Standardization, 2012).

To address this problem, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) developed ISO/IEC 29110, a quality standard that supports VSEs in improving productivity and quality at minimal cost (Laporte and O'Connor, 2017). This standard allows for efficient VSE processes and operations and increased productivity, which can be decisive when entering new markets (Laporte and O´Connor, 2016a; International Organization for Standardization, 2016). However, there is no adequate model for VSEs focused on the quality of the software product. Besides, it is complicated to implement more than one standard at the same time for this type of organization, given the restrictions associated with their specific characteristics in terms of resources, time, and budget (Laporte and O´Connor, 2017).

Within this area, a new family of standards can be highlighted, *i.e.*, ISO/IEC 25000, known as *software product quality requirements and evaluation* (SquaRE), which provides consistent terminology for specifying, measuring, and evaluating the quality of software products and systems Given the specific characteristics of VSEs, supporting their incorporation of product quality characteristics alongside the ISO/IEC 29110 process quality standard is essential.

This research focuses on identifying relevant quality characteristics and sub-characteristics in VSEs, as well as the practices they use to comply with ISO/IEC 25010 (International Organization for Standardization, 2011). product quality requirements. It will serve as a basis for implementing both standards in this type of organization.

The remainder of paper is organized as follows. The background section provides the conceptual framework for this research, including an overview of the ISO/IEC 29110 series of standards, the ISO 25000 family, and related works. Then, the materials and methods developed for data collection are described. Subsequently, the results are analyzed, and the threats to validity are discussed. The final section outlines the conclusions and potential avenues for future work.

BACKGROUND

ISO/IEC 29110 series of standards

ISO/IEC 29110 provides a series of standards and guides that feature a minimal subset of international engineering standards, known as *profiles*, tailored to the needs of VSEs (Laporte and O'Connor,

2016a, 2016b). A VSE is an entity (company, organization, department or project) composed of up to 25 individuals dedicated to software development (International Organization for Standardization, 2012). The standard targets VSEs that create non-critical products and have limited or no experience with lifecycle standards for systems engineering or software engineering, *e.g.*, the ISO/IEC/IEEE 12207 lifecycle process standard (International Organization for Standardization, 2012; Laporte and O'Connor, 2016a, 2016b).

ISO/IEC 29110 comprises a series of management and engineering guides that focus on software engineering practices for project management and software implementation, based on a set of generic profiles (International Organization for Standardization, 2012; Laporte and O'Connor, 2016a, 2016b), as shown in Figure 1.

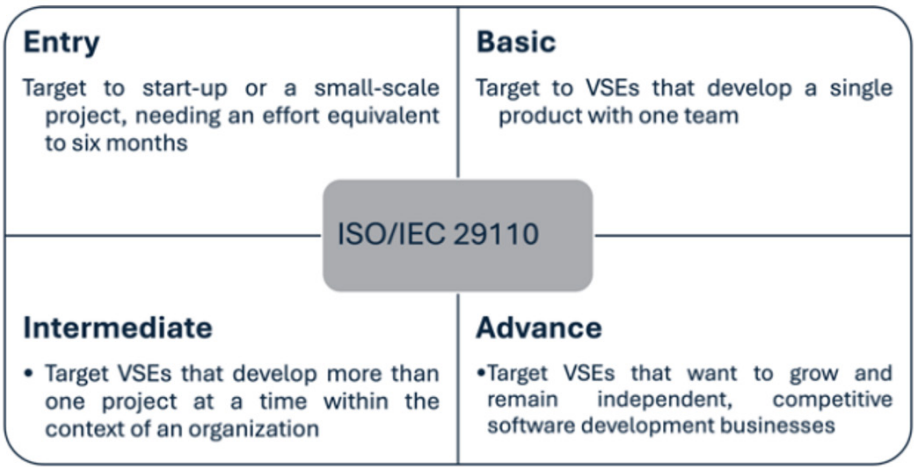


Figure 1. *Core of engineering and management profiles in ISO 29110*

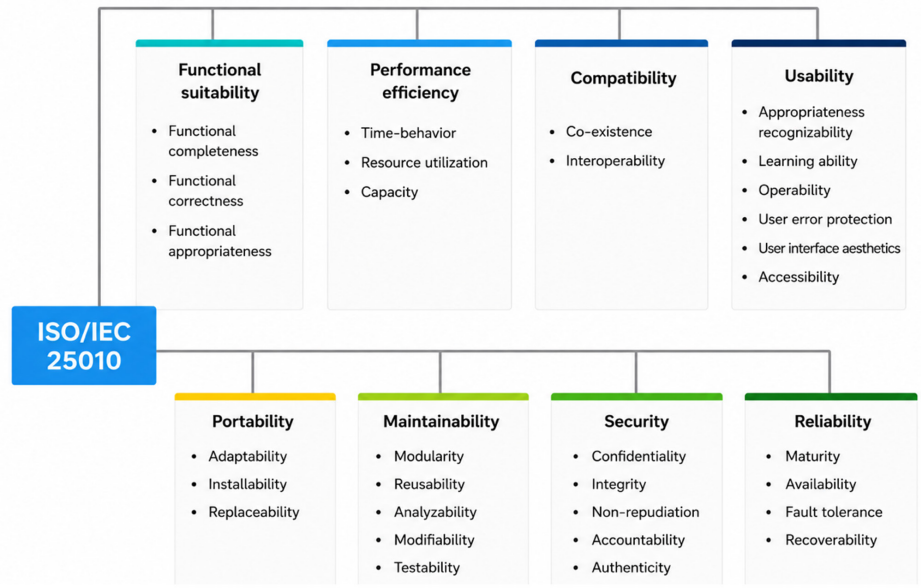
ISO 25000 family of standards

The ISO/IEC 25010 standard is a quality model for software products and quality in use. It comprises eight characteristics (subdivided into sub-characteristics) used to evaluate the static properties of software and the dynamic properties of computer systems (International Organization for Standardization, 2011).

The ISO/IEC 25000 family is composed of several parts or divisions. In this regard, the following can be highlighted: a) the standard contains various models that define the quality characteristics by which software products can be assessed (International Organization for Standardization, 2011); b) it defines software product quality metrics (International Organization for Standardization, 2019); and c) it defines the evaluation of a process, which consists of five activities that determine the tasks to be performed in order to assess software product quality (International Organization for Standardization, 2011).

As shown in Figure 2, the ISO/IEC 25010 model comprises eight quality characteristics and 31 sub-characteristics (International Organization for Standardization, 2011).

Figure 2. *ISO/IEC 25010 product quality model, adapted from ISO* (International Organization for Standardization, 2011).



Related works

This section analyzes previous research to propose improvements in software quality, focusing on both the process and the product. We categorize the research into four solution types based on our findings.

Standards harmonization

This has to do with the generation of methods that combine ISO/IEC 29110, ISO/IEC 9126, or ISO/IEC 25010 with another model, methodology, technique, practice, or other standard. We found five research works.

Solyman *et al.* (2015) propose a method that integrates the ISO/IEC 25010 standard with Level 2 of the CMMI model. This method aims to summarize the responsibilities of the customer and the supplier throughout the development lifecycle, divided into three main stages: the initial stage, the development stage, and the review and delivery stage.

Almakadmeh and Abu-Zitoon (2015) proposed an eight-step methodology based on models from the major categories of testing techniques and on the software engineer's experience. As a reference, the authors took quality characteristics such as usability, efficiency, security, reliability, maintainability, and efficiency of use, among others.

Fath-Allah *et al.* (2015) introduced a quality model for e-government portals which is grounded in the quality characteristics of ISO/IEC 25010 and best practices for these portals. The aim is to identify gaps in practice that could improve quality.

Davila and Pessoa (2015) identified the factors that most strongly influence the adoption or rejection of practices proposed in the basic profile of ISO/IEC 29110, using Hammed's information technology innovation adoption model as a reference.

García-Mireles *et al*. (2015) identified the product quality characteristics addressed in ISO/IEC 29110's basic profile for very small companies, thereby enabling the selection of practices directly related to product quality.

Software quality requirements specification assessment

This refers to generating resources—such as a method, methodology, tool, or ontology—that enable the evaluation of software requirements specification information according to ISO/IEC 29110, ISO/IEC 9126, or ISO/IEC 25010. We found five research works.

Hovorushchenko (2017) developed a method for evaluating the weights of software quality measures and indicators. This method accounts for correlations between software quality sub-characteristics. This work concludes that the presence of appropriate measures and indicators in the software requirements specification (SRS) is essential for the veracity of software quality assessments.

Raharja and Siahaan (2019) proposed a method for classifying non-functional requirements from documents using the FSKNN classifier, based on ISO/IEC 25010 categories.

Blas *et al*. (2017), proposed developing a software ontology based on the combination of three semantic models, *i.e.*, a semantic quality model (ISO/IEC 25010), metrics (the main concepts associated with the definition of a metric), and software (the content of a software product and linking quality attributes to metrics), in order to represent a software product specification.

Hovorushchenko (2018) proposed using information technology to assess the sufficiency of quality information in software requirements specifications (SRS) and ensure its accuracy through metric analysis in accordance with ISO/IEC 25010.

Software quality implementation approaches

This refers to creating a method, methodology, or tool that supports the implementation of the ISO/IEC 29110 standard, or a set of features or characteristics from ISO/IEC 9126 or ISO/IEC 25010. We identified five studies on this topic.

O'Connor and Laporte (2010) developed a series of guidelines, known as *deployment packages* (DPs), to facilitate the implementation of the ISO/IEC 29110 standard in conjunction with software development lifecycles within VSEs. DPs provide guidelines and explanations regarding the processes defined in the ISO/IEC 29110 profiles.

Muñoz *et al*. (2020) developed a five-step method for implementing ISO/IEC 29110 in VSEs which addresses training on the standard and reviews a project that uses the improved processes.

Mejía *et al*. (2019) introduced a tool to support the execution of tasks and activities within ISO/IEC 29110 software implementation and project management processes.

Moura and Santos (2018) proposed a board game designed to promote the definition of software processes based on the ISO/IEC 29110 standard (concepts related to the definition, structure, and content of a software process), where students must answer software engineering questions, define a process, and simulate one.

López *et al*. (2019) proposed selecting a set of software tools that facilitate the adoption of the ISO/IEC 29110's basic profile in VSEs. The tools suggested in Garther's quadrant were compared and selected based on cost, popularity, and available documentation.

Software product quality assessment

This refers to the evaluation of software using criteria based on ISO/IEC 9126 and ISO/IEC 25010. We found six research works.

Plöesch *et al*. (2015) presented a tool for automatically measuring maintainability-related metrics based on the IT-CISQ quality model (specifications for measuring ISO/IEC 25010 quality attributes).

Peters and Aggrey (2020) developed an analytical framework integrating the balanced scorecard (BSC) and the ISO/IEC 25010 model to assess the quality of enterprise resource planning (ERP) systems.

Virginio and Ricarte (2015) conducted a literature review to identify patient safety risks associated with electronic health records. These risks were classified based on the ISO/IEC 25010 software quality model, and they provide a basis for electronic health record (EHR) developers and regulators to address quality that may result in patient harm.

Manglapuz and Lacatan (2019) proposed evaluating the Android academic management application using the ISO/IEC 25010 model. In this study, professors, department heads, and IT experts were invited to test and evaluate the application using a customized evaluation tool.

Elcullada-Encarnación (2018) had Oman Tourism College University students evaluate the student academic advising system to determine its compliance with software quality standards. The online survey questionnaire consisted of Likert-scale items aligned with the ISO/IEC 9126-1 standard for functionality, reliability, ease of use, efficiency, maintainability, and portability.

Muniz Barros *et al*. (2015) described a methodology for evaluating software based on the NBR ISO/IEC 9126 standards (NBR ISO/IEC, 2003) and the ten golden rules for software in medical education.

After analyzing each proposal, it is evident that none addresses the quality of the unified software process and product, which justifies the elaboration of this document.

MATERIALS AND METHODS

This exploratory study aimed to identify relevant product quality characteristics in VSEs. Thus, it defines surveys as an instrument. According to Wohlin *et al*., (2012), a survey enables the collection of information from people in order to describe, compare, or explain their knowledge, attitudes, and behavior.

Survey

We implemented a survey consisting of a four-section questionnaire, as described below.

The first section requests data that may affect the relevance of quality characteristics. Specifically, it includes information such as the participant's academic degree, the role played in software projects, and the software process or product quality certification held by the participant's employer. Additionally, a specific question is used to determine whether the participant has worked in a VSE or participated in projects with fewer than 25 people, thereby assessing the validity of their response.

The second section requests data related to the type of software developed and the industrial sector to which the software product is linked. The questionnaire also covers the development methodology, programming languages, and specific practices used to meet product quality requirements.

The third section contains questions related to product quality characteristics. The quality characteristics and sub-characteristics were derived from the ISO/IEC 25010 product quality model. In total, this section includes eight quality characteristics, each with 31 sub-characteristics. For each quality subcharacteristic,

a question is presented and answered using a five-point Likert scale: not important, somewhat important, neutral, important, and very important. Moreover, this section addresses the strategies, practices, and methods implemented to meet each product quality characteristic, along with the stage of the development process at which they were introduced.

The fourth section focuses on the questionnaire, which includes four questions designed to assess its usefulness for obtaining information on product quality characteristics and for improving them. It contains questions on the suitability for identifying 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 the software project's quality requirements (Q3), and the importance of improving product quality in software projects (Q4). These four questions are answered on a five-point Likert scale: strongly disagree, disagree, neutral, agree, and strongly agree. These labels correspond to the numbers 1 to 5, in the same order. There is also an open-ended question asking the participant for suggestions on how to improve the questionnaire.

Data collection and cleaning

We used convenience sampling to collect data. This procedure involves the arbitrary selection of units without any defining criteria. The sample units are self-selected or chosen based on their availability. This method is less time-consuming and cost-effective for obtaining the sample (Mejía, 2002).

Existing collaborations between academia and industry were used to select the participant sample. The main requirements for participating in the survey were working in a VSE (with no more than 25 employees) and not developing critical software. Subsequently, the data collection procedure was conducted in two stages.

- In the first stage, we applied a supervised exploratory survey, which means that the researcher ensured that the respondents understood each question and provided a response (Kitchenham and Pfleeger, 2002). Five interviewees were surveyed, all of whom were students in the master's Program in Software Engineering at CIMAT Zacatecas. An analysis of the responses revealed a lack of clarity on how to answer the question: *"What strategies, practices, or methods were implemented in the project to ensure product quality?"* The interviewees perceived the remaining questions as useful for identifying relevant quality characteristics in software projects.
- In the second stage, the exploratory survey was conducted without supervision. We sent the survey by email to eighth- and ninth-semester students completing internships or community service in the Computer Systems Engineering program at the Technological Institute of Chalco (TESCHA), as well as to the information technology quality ecosystem of the state of Zacatecas, where the participants were involved in an ISO/IEC 29110-certified VSEs project.

We received 62 surveys. The data were then cleaned to identify incomplete, incorrect, irrelevant information, as well as other issues. The dirty data were replaced, modified, or deleted. The valid data were used for a more accurate analysis.

In summary, to clean the data, the following actions were performed:

a. One record was deleted because the answer "I don't work yet" was used in the company name field.
b. Five records were deleted from the company name field because the answer was "student".

c. One record was deleted because the answer "none" was used in the company name field.
d. There were ten empty records in the company name field. We applied a second filter to the role or position field, with the responses "student," "I do not belong to an organization," "I am not currently employed," and "I am a student," which were deemed unnecessary.
e. Four records were eliminated due to duplicate data for users.

After applying the steps presented above, we obtained a final sample of 46 valid surveys.

RESULTS

This section presents empirical evidence from the survey to identify the software quality attributes that practitioners in Very Small Entities (VSEs) consider most relevant. The results are organized into four parts: (i) participants and organization features, (ii) characteristics of the software projects, (iii) software quality characteristics and sub-characteristics perceived as relevant, and (iv) software engineering practices used to address those quality attributes.

Participants and organizations features

This section focuses on participants' profiles, including the organizations they belong to and their credentials.

1. *Enterprise*. According to the answers, 30 out of 46 respondents reported working in a VSE. The remaining participants work in SMEs.
2. *Teams*. According to the responses, most people work independently, often in pairs. Furthermore, 32 are programmers, one is a tester, three are quality assurance personnel, three are directors, two are maintenance specialists, and two are consultants.
3. *Organizational quality certification*. As reported by participants, many are unsure whether their organization holds any quality certification. This group is followed by those who know that the organization holds ISO/IEC 29110 and ISO 9000 certifications.
4. *Product quality certifications*. According to the responses, many participants are unsure whether their organization holds a quality certification for its products. However, some of them mentioned within the next most common ISO/IEC 9126 and ISO/IEC 25010 certifications.

Software projects characteristics

This section provides an overview of the type of software developed for a project. According to the answers, three types of projects were identified mobile applications, web applications and software-as-a-service, as next described:

- Most VSEs in this study develop mobile applications, followed by those dedicated to web applications. Moreover, 52% of the projects took 5-8 months to complete, 26% took 1-4 months, and 16% took 9-12 months.
- 4% were software-as-a-service projects, indicating long-lasting initiatives that depend on requirements.

Relevant quality characteristics for the participants

This section presents the set of quality characteristics and sub-characteristics that survey participants perceive as relevant. The quality characteristics and sub-characteristics were based on ISO 25010, as shown in Table 1.

Table 1. *Quality characteristics and sub-characteristics*

Quality characteristic	***Sub-characteristic***	***Importance***
Functional suitability	Functional completeness	96%
	Functional correctness	96%
	Functional appropriateness	98%
Performance efficiency	Time behavior	94%
	Resource utilization	91%
	Capacity	89%
Compatibility	Co-existence	89%
	Interoperability	83%
Interaction Capability	Appropriateness recognizability	96%
	Learnability	96%
	Operability	94%
	User error protection	89%
	User interface aesthetics	89%
	Accessibility	81%
Reliability	Faultlessness	94%
	Availability	96%
	Fault tolerance	87%
	Recoverability	87%
Security	Confidentiality	96%
	Integrity	94%
	Non-repudiation	94%
	Accountability	89%
	Authenticity	91%
Maintainability	Modularity	89%
	Reusability	85%
	Analyzability	96%
	Modifiability	98%
	Testability	87%
Flexibility	Adaptability	89%
	Installability	91%
	Replaceability	85%

As shown in Table 1, participants from VSEs consider almost all ISO/IEC 25010 quality sub-characteristics relevant to their software projects, although their perceived importance varies. The highest-rated sub-characteristics, with 98%, were functional appropriateness and modifiability, followed by functional completeness, functional correctness, learnability, availability, confidentiality, and analyzability, with 96%. The findings highlight that VSEs associate software quality with delivering correct functionality and efficiently evolving software.

Subcharacteristics such as accessibility, interoperability, replaceability, reusability, recoverability, fault tolerance, and testability received lower importance ratings (81%-87%), which could indicate that they are not perceived as having the same level of criticality in their projects.

Practices used to address product quality

This section presents the practices participants used to meet the quality characteristics. Tables 2 to 9 present the list of quality characteristics and practices. Additionally, we present an analysis of the practices for which at least two answers are provided.

Table 2 presents the practices related to functional suitability. Out of the 46 surveys, 17 participants reported 18 practices, 15 provided information but did not specify any practice that contributed to the study, and the remaining 13 did not provide any information.

Table 2. *Practices for achieving functional suitability*

Received answers	***Reported practices***	***Frequency***
18	Developing functional tests	4
	Developing unit tests	2
	Implementing security standards	1
	Implementing development methodologies	1
	Implementing quality models	1
	Implementing tests	3
	Implementing traceability records	2
	Verifying and validating software requirements	4

As shown in the Table, functional tests and verification and validation of requirements were the practices most frequently associated with addressing functional suitability. Then, they conduct testing to ensure that the implemented functionality meets stakeholder requirements.

Table 3 presents the practices related to performance efficiency. As the shown in the Table, of the 46 surveys, 15 participants reported 19 practices, and 17 provided information. Still, they did not specify any practice that contributed to the study, and the remaining 14 provided no information.

Table 3. *Practices for achieving performance efficiency*

Received answers	*Reported practices*	*Frequency*
	Developing integration tests	1
	Developing beta tests	1
	Developing stress tests	1
	Developing performance tests	2
	Developing systems tests	1
	Establishing thresholds	1
	Using frameworks	1
19	Implementing optimized algorithms	1
	Implementing development methodologies	1
	Implementing monitoring and control	1
	Implementing software architecture patterns	1
	Implementing tests	4
	Implementing software reengineering	1
	Performing software requirements specification	1
	Refactoring code	1

As shown in the Table, participants focus on implementing software testing and optimization of code practices to address performance efficiency.

Table 4 presents the identified compatibility-related practices. Out of the 46 surveys, 14 participants reported 15 practices, and 17 provided information. Still, they did not specify any practice that contributed to the study. The remaining 15 provided no information.

Table 4. *Practices for achieving compatibility*

Received answers	*Reported practices*	*Frequency*
	Implementing hybrid architecture systems	1
	Implementing coding standards	2
	Implementing security standards	1
	Implementing communication standards	1
15	Implementing tests	2
	Implementing compatibility tests	2
	Implementing a benchmarking study	1
	Implementing a compatibility strategy	4
	Verifying software requirements	1

As shown in the Table, practitioners focus on using compatibility strategies, coding standards, and compatibility tests to ensure consistent behavior.

Table 5 presents the practices related to interaction capability. Out of the 46 surveys, 17 participants reported 23 practices, 14 provided information but did not specify any practice that contributed to the study, and the remaining 15 did not provide any information.

Table 5. *Practices for achieving interaction capability*

Received answers	*Reported practices*	*Frequency*
23	Developing user interface designs	3
	Developing beta tests	1
	Developing usability tests	2
	Developing unit tests	1
	Developing a navigation model	1
	Implementing role-based access control	2
	Implementing cascading style sheets	2
	Implementing user interface (UI) design patterns	1
	Implementing tests	3
	Implementing user-centered design	2
	Creating a user manual	1
	Performing design validations	3
	Creating video tutorials	1

As shown in the Table, practitioners focus on interface design and design validation to achieve interaction capability, along with test implementation.

Table 6 presents the compatibility-related practices. Out of the 46 surveys, 15 participants reported 18 practices, 13 provided information but did not specify any practice that contributed to the study, and the remaining 18 provided no information.

Table 6. *Practices for achieving reliability*

Received answers	*Reported practices*	*Frequency*
18	Contracting cloud services according to project requirements	1
	Developing a risk management plan	1
	Develop a project plan	1
	Implementing redundant components	1
	Implementing transaction control commands	1
	Implementing server monitoring	1
	Implementing recovery testing	1
	Implementing tests	5
	Implementing fault tolerance techniques	1
	Implementing unit tests	1
	Implementing data server replication	1
	Keeping online information up to date	1
	Performing server backups	1
	Verifying software requirements	1

As shown in the Table, practitioners associated the achievement of reliability with the implementation of tests.

Table 7 presents the security-related practices. Out of the 46 surveys, 15 participants reported 24 practices, 14 provided information but did not specify any practice that contributed to the study, and the remaining 17 provided no information.

Table 7. *Practices for achieving security*

Received answers	***Reported practices***	***Frequency***
24	Establishing a user account lockout policy	1
	Establishing a password expiration policy	1
	Implementing role-based access control	5
	Implementing user action logging	2
	Implementing a network firewall	1
	Implementing authentication methods	3
	Implementing encryption algorithms	2
	Implementing security modules using frameworks	1
	Implementing software usage monitoring through tools	1
	Implementing HTTPS protocols	2
	Implementing tests	2
	Implementing penetration testing	1
	Performing database encryption	1
	Backing up the database	1

As shown in the Table, practitioners achieve security primarily through role-based access control, followed by authentication methods that prioritize access management as a secure mechanism.

Table 8 presents the practices related to maintainability. Out of the 46 surveys, 18 participants reported 26 practices, and 12 provided information. Still, they did not specify any specific practice that contributed to the study. The remaining 16 did not provide any information.

Table 8. *Practices for achieving maintainability*

Received answers	*Reported practices*	*Frequency*
26	Defining a maintenance organizational structure	1
	Developing a maintenance manual	2
	Developing an operation manual	2
	Using frameworks	1
	Implementing software updates	2
	Implementing MVC architectures	1
	Implementing software change control	5
	Implementing continuous integration	1
	Implementing modular software programming	3
	Implementing preventive and corrective maintenance for hardware and software	1
	Implementing pipelines	1
	Implementing tests	1
	Implementing branch testing	1
	Implementing software version control	3
	Backing up the source code	1

As shown in the Table, practitioners prefer implementing software change control to achieve maintainability, followed by implementing modular software programming and software version control, which allow them to evolve software under control.

Table 9 presents the practices related to flexibility. Out of the 46 surveys, 13 participants reported 15 practices, 17 provided information but did not specify any practice that contributed to the study, and 16 provided no information.

Table 9. *Practices for achieving flexibility*

Received answers	*Reported practices*	*Frequency*
15	Developing a maintenance manual	1
	Developing an operations manual	1
	Developing software in interpreted languages	1
	Developing cross-platform software	2
	Implementing instances	1
	Implementing MVC architectures	1
	Implementing a software container	1
	Implementing high-level programming	1
	Implementing tests	1
	Backing up the source code	2
	Refactoring code	1
	Using software from the same vendor	2

As shown in the Table, practitioners prefer implementing three practices: developing cross-platform software, backing up the source code, and using software from the same vendor to achieve flexibility.

In summary, the practices implemented by users tend to be limited in scope and address multiple quality characteristics, with software testing being the most commonly used. Besides, there is a perception of limited use of more formal practices in quality process management that can be associated with their capabilities.

THREATS TO VALIDITY

Threats to validity are important considerations in the design, administration, and documentation of a survey, as their presence can diminish the validity of the results, complicate interpretation, and lead to erroneous conclusions and inferences (Carrillo *et al.*, 2020).

The exploratory survey provided information on the product quality characteristics and sub-characteristics that VSEs consider in their software development projects, and it aimed to identify the practices they employ to meet these characteristics. Although the necessary guidelines for conducting surveys were followed (Kitcheham and Pfleeger, 2002), this type of empirical research has several limitations.

This was an exploratory study, and the participants were selected via convenience sampling. On the one hand, 24 students enrolled in professional internships during their eighth and ninth semesters at TESCHA were selected. This level was chosen because students undertake theoretical-practical projects in which they solve specific problems related to the social and productive realities of companies, organizations, or government agencies. On the other hand, 22 enterprises (VSEs) were included, ten of which belong to the Zacatecas quality ecosystem. Consequently, the list of the most relevant characteristics, as well as the practices implemented to achieve them, cannot be statistically generalized to other enterprises. Still, the results provide a starting point for formulating appropriate hypotheses regarding software product quality in enterprises.

CONCLUSIONS

This paper presents the results of using a survey instrument to collect data on product quality characteristics and practices in VSEs. The survey enabled the collection of information on product quality characteristics and practices implemented in VSEs

The sample analyzed in this paper comprised 46 responses, which may have compromised the validity of the results. However, the results are valuable because we did not find prior work that examined VSEs and ISO/IEC 25010 in Mexico or across Latin America.

The exploratory analysis enabled the identification of relevant product quality characteristics in the VSEs. The results showed that all 31 sub-characteristics are essential, with values ranging between 81 and 98% across all participants.

Additionally, the exploratory analysis identifies the practices that VSEs implement to address product quality characteristics. The study of the practices reported by participants revealed insufficient information on compatibility, performance efficiency, and portability, with only six, seven, and eight practices reported, respectively.

On the other hand, the characteristics with the highest number of reported practices are security and maintainability, with 16 and 17, respectively. Moreover, the results suggest that these aspects are the most relevant characteristics for VSEs.

As further work, the authors are currently tracing each product quality characteristic using the ISO/IEC 29110 standard, in order to identify the activities and/or work products in which they should be implemented within Project Management and Software Implementation processes.

ACKNOWLEDGEMENTS

We would like to thank the students who participated in the survey, as well as the Center for Mathematics Research (CIMAT), Universidad de Sonora (UNISON), and SECIHTI for their support in performing this research.

AUTHOR CONTRIBUTIONS

Mirna Muñoz: conceptualization, methodology, supervision, writing
Gabriel Alberto García-Mireles: conceptualization, methodology, supervision, review
Jezreel Mejía: supervision, writing (review and editing)
Elizabeth Villanueva: research, data curation, formal analysis, writing
All authors have read and agreed to the published version of the manuscript.

DECLARATION OF THE USE OF ARTIFICIAL INTELLIGENCE

The authors use Grammarly to correct writing.

REFERENCES

Almakadmeh, K., and Abu-Zitoon, F. (2015). A generic model-based methodology of testing techniques to obtain high quality software. IN ACM (Eds.), *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15)* (article 49). Association for Computing Machinery. https://doi.org/10.1145/2816839.2816903

Associação Brasileira de Normas Técnicas (ABNT) (2003). *NBR ISO/IEC 9126-1. Engenharia de software: qualidade de produto: parte 1: modelo de qualidade*. ABNT.

Blas, M.J., Gonnet, S., and Leone, H. (2017). An ontology to document a quality scheme specification of a software product. *Expert Systems*, *34*(5), e12213. https://doi.org/10.1111/exsy.12213

Carrillo Avalos, B., Sánchez Mendiola, M., and Leenen, I. (2020). Amenazas a la validez en evaluación: implicaciones en educación médica. *Investigación en Educación Médica*, *9*(34), 100-107. https://doi.org/10.22201/facmed.20075057e.2020.34.221.

Dávila, A., and Pessoa, M. (2015). Factors driving the adoption of ISO/IEC 29110: A case study of a small software enterprise. In IEEE (Eds.), *Proceedings of the 2015 Latin American Computing Conference (CLEI)* (pp. 1-8). https://doi.org/10.1109/CLEI.2015.7360042

Elcullada-Encarnación R. F. (2018). *Academic advising system using data mining method for decision making support* [Conference article]. 4th International Conference on Computer and Technology Applications (ICCTA). https://doi.org/10.1109/CATA.2018.8398651

Fath-Allah, A., Cheikhi, L., Al-Qutaish, R. E., and Idri, A. (2015). E-Government portals maturity models: A best practices' coverage perspective. *Journal of Software*, 10(7), 805-824. https://www.jsoftware.us/vol10/66-JSW1510.pdf

García-Mireles, G. A., de la Rubia, M. Á. M., García, F., and Piattini, M. (2015). *Identifying quality characteristic interactions during software development* [Conference article]. Proceedings of the International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE).

Hovorushchenko, T. (2017). Method of evaluating the weights of software quality measures and indicators. *Application and Theory of Computer Technology*, *2*(2), 16-25. https://ceur-ws.org/Vol-2104/paper_228.pdf

Hovorushchenko, T. O. (2018). Methodology of evaluating the sufficiency of information for software quality assessment according to ISO 25010. *Journal of Information and Organizational Sciences*, *42*(1), 4. https://doi.org/10.31341/jios.42.1.4

International Organization for Standardization. (2011). *ISO/IEC 25040:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*.: https://www.iso.org/standard/35733.html

International Organization for Standardization (2012). *ISO/IEC TR 29110-5-1-2:2012, Software Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-1-1: Management and Engineering Guide — Entry Profile*. International Organization for Standardization (ISO). http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

International Organization for Standardization (2016). *ISO/IEC TR 29110-1:2016 Systems and software engineering—Lifecycle profiles for Very Small Entities (VSEs)—Part 1: Overview*. https://www.iso.org/standard/62711.html

International Organization for Standardization (2019). *ISO/IEC 25020:2019 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework*. https://www.iso.org/standard/72117.html

Kitchenham, B. A., and Pfleeger, S. L. (2002). Principles of survey research part 2. *ACM SIGSOFT Software Engineering Notes*, 27(1), 18-20. https://doi.org/10.1145/566493.566495

Laporte, C. Y., and O'Connor, R. V. (2016). Systems and software engineering standards for very small entities: Accomplishments and overview. *Computer,* 49(8), 84-87. https://doi.org/10.1109/MC.2016.242.

Laporte, C. Y., and O'Connor, R.V. (2016b) Software process improvement in industry in a graduate software engineering curriculum. *Software Quality Professional Journal, ASQ, 18*(3), 4-17. https://ceur-ws.org/Vol-1368/paper3.pdf

Laporte, C. Y., and O'Connor, R. V. (2017). Software process improvement standards and guides for very small organizations: An overview of eight implementations. *CrossTalk: The Journal of Defense Software Engineering, 30*(3), 23-27. https://doras.dcu.ie/21798/1/Published.pdf

López, V. M. (2018). La competitividad de las PYMES en México: retos y oportunidades ante un mundo globalizado. *Horizontes de la Contaduría en las Ciencias Sociales, 5*(9), 79-91. https://www.uv.mx/iic/files/2018/12/num09-art07-105.pdf

Lopez, W., Bernuy, W., and Pando, B. (2019). Bringing a micro enterprise towards the ISO 29110. In IEEE (Eds.), *Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). IEEE. https://doi.org/10.23919/CISTI.2019.8760865

Manglapuz S. J. R., and Lacatan, L. L. (2019). Academic management android application for student performance analytics: A comprehensive evaluation using ISO 25010:2011. *International Journal of Innovative Technology and Exploring Engineering*, *8*(12), 5085-5089. https://doi.org/10.35940/ijitee.l2735.1081219

Mejía, J. (2002). *Problemas metodológicos de las ciencias sociales en el Perú*. Fondo Editorial de la UNMSM.

Mejía, J., Muñoz, M., Montoya-Méndez, P., Girón-Bobadilla, H., and Rodríguez-Maldonado, I. (2019). *ISO/IEC 29110 implementation tools proposal (basic profile)* [Conference article]. *2019 8th International Conference on Software Process Improvement (CIMPS)*, Leon, Mexico. https://doi.org/10.1109/CIMPS49236.2019.9082430

Moura, V., and Santos, G. (2018). ProcSoft: A board game to teach software processes based on ISO/IEC 29110 standard. In ACM (Eds.), *Proceedings of the XVII Brazilian Symposium on Software Quality (SBQS '18)* (pp. *363-372*. Association for Computing Machinery. https://doi.org/10.1145/3275245.3276319

Muniz Barros, P. R., Cazella, S. C., and Flores, C. D. (2015). *Analyzing softwares in medical education focusing on quality standards* [Conference article]. 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, Sao Carlos, Brazil. https://doi.org/10.1109/CBMS.2015.20

Muñoz, M., Mejía, J., and Laporte, C.Y. (2020), Implementing ISO/IEC 29110 to reinforce four very small entities of Mexico under an agile approach. *IET Software, 14*(2), 75-81. https://doi.org/10.1049/iet-sen.2019.0040

Mellado Fernández, D., Rodríguez, M., Verdugo, J., Piattini, M., and Fernández-Medina, E. (2010). *Evaluación de la calidad y seguridad en productos software*. https://administracionelectronica.gob.es/pae_Home/dam/jcr:4bda2362-75fb-4668-9437-0afd55aad0e5/28inciativas-legales.pdfandved=2ahUKEwjfjof0xMGVAxUple4BHQ1gHIQQFnoECB4QAQandusg=AOvVaw201QBVRsMI55m-NeR_BEJq

O'Connor, R. V., and Laporte, C. Y. (2010). Towards the provision of assistance for very small entities in deploying software lifecycle standards. En *Proceedings of the 11th International Conference on Product-Focused Software Development and Process Improvement (PROFES 2010)*. ACM. https://doi.org/10.1145/1961258.1961259

Peters, E., and Aggrey G. K. (2020). An ISO 25010 based quality model for ERP systems. *Advances in Science, Technology and Engineering Systems Journal*, *5*(2), pp. 578-583. https://doi.org/10.25046/aj050272

Plöesch R., Schuerz, S., and Koerner, C. (2015). *On the validity of the IT-CISQ quality model for automatic measurement of maintainability* [Conference article]. 2015 IEEE 39th Annual Computer Software and Applications Conference, Taichung, Taiwan. https://doi.org/10.1109/COMPSAC.2015.47

Raharja, I. M. S., and Siahaan, D. O. (2019). *Classification of non-functional requirements using fuzzy similarity KNN based on ISO/IEC 25010* [Conference article]. 2019 12th International Conference on Information and Communication Technology and System (ICTS), Surabaya, Indonesia. https://doi.org/10.1109/ICTS.2019.8850944

Sánchez-Gordon, M. L., O'Connor, R. V., and Colomo-Palacios, R. (2015). Evaluating VSEs viewpoint and sentiment towards the ISO/IEC 29110 standard: A two country grounded theory study. In T. Rout, R. O'Connor, and A. Dorling, A. (Eds.), *Software Process Improvement and Capability Determination*. *SPICE* 2015 (pp. 114-127). Springer. https://doi.org/10.1007/978-3-319-19860-6_10

Solyman, A. M., Ibrahim, O. A., and Elhag, A. A. M. (2015). *Project management and software quality control method for small and medium enterprise* [Conference article]. 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), Khartoum, Sudan. https://doi.org/10.1109/ICCNEEE.2015.7381442

Virginio, L. A., and Ricarte, I. L. (2015). Identification of patient safety risks associated with electronic health records: A software quality perspective. *Studies in Health Technology and Informatics, 216*, 55-59. https://pubmed.ncbi.nlm.nih.gov/26262009/

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