Requirements Engineering (2023) 28:257–281
https://doi.org/10.1007/s00766-022-00394-4

ORIGINAL ARTICLE

Measuring and improving software requirements elicitation in a small-sized software organization: a lightweight implementation of ISO/IEC/IEEE 15939:2017—systems and software engineering—measurement process

C. Pacheco[1] · I. Garcia[1] · J. A. Calvo-Manzano[2] · M. Reyes[1]

Received: 22 October 2021 / Accepted: 9 November 2022 / Published online: 19 November 2022

Abstract
Requirements engineering is one of the most important areas of software engineering because it enables practitioners to determine the characteristics and constraints of the software to be developed. Therefore, the requirements elicitation process aims to obtain a preliminary version of the requirements before their specification, but it is difficult to know if the elicited requirements have the desired attributes (i.e., these requirements are complete, correct, consistent, etc.) due to the nature of this process. There are some measures that allow requirements engineers to quantify certain aspects of the requirements elicitation process and its final product (i.e., the preliminary version of the requirements, also known as preliminary requirements), but there is no way of measuring and controlling the quality of the elicited preliminary requirements. With the aim of understanding, evaluating, and improving the activities of the requirements elicitation process, some of the existing measures for this process were analyzed and, consequently, a case study was conducted to define and implement a measurement program with a set of eight measures that are proposed in this study. This program is based on the ISO/IEC/IEEE 15939:2017—systems and software engineering—measurement process, an international standard that establishes a common process and framework for the measurement of systems and software. The measurement program corresponds to a lightweight implementation of the standard in the context of a small-sized software organization (where the size of the software staff is 11–50 people, the size of projects is 50,000–100,000 LOC, and the time spent on the projects is six to twelve months) and the obtained results showed a preliminary positive influence when obtaining high-quality preliminary requirements.

Keywords ISO/IEC/IEEE 15939:2017 · Requirements elicitation process · Measurement program · Measures · Small-sized software organization

1 Introduction

In the context of the software development process, requirements engineering (RE) is one of the most important areas that helps software practitioners to decide and define what the software must do [1]. Kotonya and Sommerville [2] and Berenbach et al. [3] stated that RE is frequently described by the following processes: elicitation, analysis, specification, verification and validation, and management of a set of software requirements. Therefore, the final product of the RE processes is the software requirements specification (SRS) which establishes the functional and non-functional requirements of the software that will be developed. However, before obtaining a SRS, it is necessary to obtain a list of preliminary requirements from the stakeholders by performing the requirements elicitation process.

A preliminary requirement includes, for example, the stakeholder needs, system concepts, user expectations, and the system environment (e.g., "able to specify that the language function checks for correct spelling," "runs under

✉ I. Garcia
ivan@mixteco.utm.mx; leninca@mixteco.utm.mx; karenina@mixteco.utm.mx

J. A. Calvo-Manzano
joseantonio.calvomanzano@upm.es

[1] División de Estudios de Posgrado, Universidad Tecnológica de La Mixteca, Huajuapan de León, Mexico

[2] Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain

Linux" and "does not require conversion of existing files"). In this regard, eliciting "good" preliminary requirements is crucial in ensuring that the design and the development of the software are based on the right requirements [4–6]. However, the requirements elicitation process is not straightforward and usually cumbersome due to multiple resources with different perspectives.

According to Bourque and Fairley [7], Sandhu and Weistroffer [8], Dar et al. [9], and the ISO/IEC/IEEE 29148:2018 standard [10], the requirements elicitation process is frequently defined by three main activities: (1) *identifying the individual stakeholders or stakeholder classes* (i.e., the people who provide the information needed in order to identify the problem that needs solving, as well as their needs and constraints); (2) *eliciting stakeholder requirements from the identified stakeholders* (i.e., one list of preliminary requirements per stakeholder); and (3) *integrating, refining, and organizing the collected information*[1] (i.e., the list of preliminary requirements) in order to determine the functionalities and limitations of the desired software.

With the aim of obtaining better results from the requirements elicitation process, a measurement activity may be needed to improve and control the end product (i.e., the final list of preliminary requirements) and provide positive feedback to all stakeholders (e.g., programmers, project managers, customers, etc.) [11, 12]. However, current literature only provides empirical evidence for measurement efforts concerning activity 2 of the requirements elicitation process (i.e., *eliciting stakeholder requirements from the identified stakeholders*) [13–20].

Therefore, this paper focus on adapting the ISO/IEC/IEEE 15939:2017 to implement a measurement program for activities 1 and 3 of the requirements elicitation process (i.e., *identifying the individual stakeholders or stakeholder classes*; and *integrating, refining, and organizing the collected information*) with the aim of reducing development team effort and increasing the quality of the preliminary requirements in a small-sized software organization. This standard describes a measurement process applicable to system and software engineering as well as management disciplines but there are no guidelines for its adaptation in a small context as yet.

In summary, a measurement program was implemented to measure and improve the other two activities (1 and 3) neglected to this day: stakeholder identification activity, and verifying the quality of the final list of the elicited preliminary requirements (i.e., if the list is complete, unambiguous, consistent, traceable, and understandable). Our approach only focuses on covering five of the nine quality attributes of individual requirements proposed by Lauesen [21] and the ISO/IEC/IEEE 29148:2018 [10] (i.e., unambiguous, consistent, traceable, and understandable preliminary requirements, and the completeness of the preliminary requirements list), because the other four (i.e., correct, ranked for importance and stability, modifiable, and verifiable) are applied during the requirements analysis process [22, 23].

The rest of the paper is organized as follows. Section 2 provides the background of the research on ISO/IEC/IEEE 15939:2017 and previous efforts to measure the requirements elicitation process. Section 3 describes a case study regarding to the application of the ISO/IEC/IEEE 15939:2017 into a small-sized software organization and presents the main threats for validating the achieved results. Finally, Sect. 4 summarizes the main conclusions of this study and some future work.

2 Background

An airplane pilot needs numerous instruments when measuring different parameters to make sure that everything is working correctly in order to operate a flight successfully from one point to another. According to Kneuper [24], there is a certain similarity between an airplane and the software process, as the latter must also undergo validation to investigate whether it is successful in meeting its specifications. In this context, a measurement program can provide a useful basis for such validation, providing timely information about the performance and quality of software processes, and thereby helping practitioners to identify shortcomings and opportunities for improvement. Tahir et al. [25] stated that a software measurement program is a crucial tool for understanding, evaluating, managing, and improving software processes, products, and/or resources. Therefore, measuring the software process and its outcomes enables organizations to gain knowledge about them. Many software organizations can use measurement to better understand their software processes so that their performance and quality can be predicted and improved. Consequently, these organizations can make timely appropriate and informed decisions during the software development process.

As previously mentioned, RE processes have a significant impact on the successful completion of software projects. Therefore, it is important to elicit good requirements to carry out a thorough and proper requirements elicitation process. According to Nordin et al. [26], measuring the quality of requirements is the key to achieving this goal and selecting high quality requirements. Moreover, this quality will contribute to the successful and cost-effective creation of software that solves real user needs. Research by Ahmad

[1] This activity corresponds to the activity "1. Analyze the complete set of elicited requirements" defined by section "6.2.3.3 Analyze and maintain stakeholder requirements" from the ISO/IEC/IEEE 29148:2018 standard.

and Asmai [12] stated, for example, that this measurement could be done at a very early stage of RE, specifically in the requirements elicitation process. With this aim in mind, public and not-for-profit organizations, including software organizations, could achieve significant benefits by achieving and maintaining the competitive growth of their products from a market-driven perspective. However, the implementation of measurement programs is still a considerable challenge for many software organizations, even more so for small-sized ones. These organizations usually must adapt certain models or standards such as ISO/IEC 14598-x [27], the ISO/IEC 9126 standard family [28], ISO/IEC/IEEE 15939:2007 [29], CMMI-DEV [30], or ISO/IEC 25021 [31], using mainly past experiences or guidelines generated by domain experts.

This study will now focus on contributing to this objective by conducting a case study to define and implement a measurement program for the requirements elicitation process in the context of a small-sized software organization.

2.1 Previous experiences of measuring the requirements elicitation activities

Planning and implementing a measurement program are quite complicated regardless of the software organization's size. Nevertheless, some organizations have focused their efforts on measuring the RE processes to improve their quality and chances of success for the software development life cycle.

Research by Palyagar [32], for example, proposed some high and low-level measures for RE processes quality and presented a framework that identifies factors that can have an impact on quality. The framework can be tailored based on organization and project specifics to then link high and low-level process measurements to requirements quality using a decision matrix. Using this method, basic, underlying, and immediate factors that may be influencing RE processes quality are identified. The results from a case study demonstrated that measuring RE processes quality and identifying factors that influence process quality can help make RE and its benefits more relevant and tangible to organizations.

Similarly, Niazi [33] proposed a measuring instrument to measure the maturity of the RE processes and, consequently, develop better ways of assisting practitioners in effectively measuring maturity. The main objective of this research was to provide software organizations with a clear idea of which RE processes are weak and require further consideration. The instrument was based on Sommerville's requirements maturity model and was tested in a case study where the requirements elicitation process was measured. The obtained results showed that participants effectively judged the effectiveness of different requirements elicitation practices. A few years later, Niazi's [34] research evolved and a RE maturity measurement framework was developed for assessing the maturity of RE processes. The main purpose of this framework was to assist practitioners with measuring the maturity of the RE processes within organizations more efficiently. The results obtained from two case studies showed potential in demonstrating that the practitioners were able to effectively judge the strengths and weaknesses of the RE processes.

Furthermore, Génova et al. [35] presented a few low-level indicators for measuring quality in textual requirements, as well as a tool that computes quality measures in a fully automated way. Therefore, the aim of the research was to emphasize the need for measuring to point out obvious defects and provide suggestions for improving the quality of the requirements. In this regard, efforts were focused on the textual quality analysis of requirements, using formal requirements documents as input information. The study was not directly concerned with the meaning or content of requirements, nor with the process to elicit the right requirements. The results showed that low-level quality indicators can provide valuable insights for improving high-level quality features of requirements.

Research by Bhatti et al. [36] introduced a framework to help elicitors validate the requirement elicitation process using prioritization techniques to effectively select the appropriate elicitation technique. The requirements elicitation framework was comprised of four stages: identification of the domain, management of stakeholders, selection of the requirement elicitation techniques and selection of the requirements prioritization techniques. Moreover, a set of quality measures and estimations was defined to validate elicited functional and non-functional requirements. A case study was conducted to test the framework, and three main issues were identified: the need for the integration of functional requirements with non-functional requirements, the relevance of solving all the conflicting requirements, and the importance of avoiding ambiguous specifications.

Finally, Pinto et al. [37] presented an agile quality framework focused on evaluating the quality regarding the management of requirements and specifications in real production environments on agile projects. Moreover, the framework facilitated the evaluation of quality in agile environments with the development process considered as the object for measurement, regardless of the agile focus selected. During the validation experience in real environments, it was possible to verify that the framework offered development teams a powerful, easy-to-use, and reliable tool when measuring the quality associated with requirements management.

With the aim of highlighting the relevance of each aforementioned study, Table 1 shows the main contribution of each study and the associations that can be made with our research.

Table 1 Studies related to measuring the requirements elicitation activities within real contexts

Study	Coverage of requirements elicitation activities	Used measurement framework (model or standard)	Context	Know-how from the measurement program
Palyagar [32]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders	Not specified	One large-sized software organization	Not provided
Niazi [33]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders	No framework was applied, only an evaluation instrument was designed to collect information and assess the maturity of the elicited requirements	Two medium-sized software organizations	Not provided
Niazi [34]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders	No framework was applied. An evaluation instrument was designed to collect information and assess the maturity of the elicited requirements	Two large-sized software organizations	Not provided
Génova et al. [35]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders	No framework was applied. A software tool was developed to collect information on elicited requirements through DOORS	The tool was commercialized in Europe for large-sized software organizations	Not provided
Bhatti et al. [36]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders	Not specified. Quality measures (e.g., reliability, usability, efficiency) were evaluated for non-functional-requirements	One large-sized software enterprise	Not provided
Pinto et al. [37]	Activity 2. Eliciting stakeholder requirements from the identified stakeholders (partially, because the quality of the collected user stories is assessed)	The Agile Quality Model and the QuAGI software tool were introduced to measure the quality of user stories	Not specified	Not provided
This study	Activity 1. Identifying the individual stakeholders or stakeholder classes, and Activity 3. Integrating, refining, and organizing the collected information	ISO/IEC/IEEE 15939:2017	One small-sized software enterprise	All the information related to the design and management of the measurement program is provided

As can be seen in these previous works, efforts have been made to measure the requirements elicitation activities to improve the requirements quality within software organizations, mainly focusing on activity 2 from the requirements elicitation process (i.e., *eliciting stakeholder requirements from the identified stakeholders*). However, this paper concentrated on adapting the ISO/IEC/IEEE 15939:2017 to implement a measurement program for activities 1 and 3 of the requirements elicitation process (i.e., *identifying the individual stakeholders or stakeholder classes*; and *integrating, refining, and organizing the collected information*). Moreover, the measurement program is contextualized for a small-sized software organization. It is worth mentioning that there was no intention to provide adaptation guidelines for small-sized software organizations, as this would require conducting additional case studies and refining the activities and tools proposed in this study to facilitate the understanding of the ISO/IEC/IEEE 15939:2017 in a particular organization. On the other hand, we are presenting all the pertinent information related to the experience derived from an adaptation of this standard into a small-sized software organization with the aim of generating knowledge that could be reused in later adaptations. Therefore, the following section presents how the standard was adapted to accomplish this objective.

3 Research methodology

Within the software measurement domain, there are many models, standards, tools, and practices for collecting and analyzing measurement data for improving software processes, products, and/or resources. A few of them were created considering the characteristics of small-sized software organizations. However, implementing a measurement program is a more difficult task for these organizations due their culture, limited human resources, limited training, and poor software measurement knowledge [38]. Bearing in mind these issues, it is true to state that measurement programs are more difficult to perform in small-sized software organizations than in large-sized software organizations [39, 40].

We therefore believe that adapting an existing measurement standard could simplify the measurement process for these types of organizations. Therefore, efforts were concentrated on the two core measurement activities of the ISO/IEC/IEEE 15939:2017: the *Prepare for Measurement* activity and the *Perform Measurement* activity, as detailed in the following sections.

3.1 ISO/IEC/IEEE 15939:2017 and the software measurement programs

Any software process standard defines a technical specification that has been previously reviewed and approved by different experts to reach a consensus in prescribing a disciplined approach for specific areas of software engineering. In the software measurement context, for example, the ISO/IEC/IEEE 15939:2017 [41] provides the definition of a measurement process applicable to system and software engineering and management disciplines. According to the documentation of the ISO/IEC/IEEE 15939:2017, the process is described through a model that defines the measurement process activities that are required to appropriately specify which measurement information is required, how the measures and analysis results are to be applied, and how to determine if the analysis results are valid. Therefore, this standard describes the process, activities, and tasks that are needed to successfully identify, define, select, apply, and improve measurements within an overall project or organizational measurement structure.

The ISO/IEC/IEEE 15939:2017 is intended for use by (a) suppliers, including personnel performing management, technical and quality management functions in system and software development, maintenance, integration, and product support organizations, and (b) acquirers, including personnel performing management, technical and quality management functions in procurement and user organizations. Furthermore, the ISO/IEC/IEEE 15939:2017 establishes a common process and framework for the measurement of systems and software. This measurement process can be applied to the project and products across the life cycle to aid the planning, managing, assessing, and decision-making at all stages of a system or software life cycle. The standard also provides activities for supporting the definition, control and improvement of the measurement process used within an organization or a project. Moreover, the users of the standard are responsible for selecting a set of measures for the project and defining the application of those measures across the process, products, and other elements of the life cycle. These users are also responsible for selecting and applying appropriate methods, models, tools, and techniques considered suitable for the project. Similarly, the standard is not intended for prescribing the name, format, explicit content, or recording media of the information items to be produced. Rather, the measurement process establishes a set of objectives (called "outcomes") along with a set of activities and tasks to achieve those objectives. Users who implement the activities and tasks can assert full conformance to tasks. Some users, however, might have innovative measurement process variants that achieve the objectives (i.e., the outcomes) of the declared set of processes without implementing all the activities and tasks. Furthermore, the order in which activities are presented does not necessarily imply an order of implementation.

The measurement process consists of four activities that are sequenced in an iterative cycle allowing for continuous feedback and improvement. Two activities are considered

Table 2 ISO/IEC/IEEE 15939:2017 process

Activities	Tasks
Establish and sustain measurement commitment	T1. Define the measurement requirements and scope T2. Establish the management commitment T3. Assign resources for the measurement activities
Prepare for measurement	T4. Define the measurement strategy T5. Describe the characteristics of the organization that are relevant to measurement T6. Identify and prioritize the information needs T7. Select and specify measures that satisfy the information needs T8. Define data collection, analysis, access, and reporting procedures T9. Define criteria for evaluating the information items and the measurement process T10. Identify and plan for the enabling systems or services to be used T11. Review, approve, and provide resources for measurement tasks T12. Acquire and deploy supporting technologies
Perform measurement	T13. Integrate procedures for data generation, collection, analysis, and reporting into the relevant processes T14. Collect, store, and verify data T15. Analyze data and develop information items T16. Record results and inform the measurement users
Evaluate measurement	T17. Evaluate information products and the measurement process T18. Identify potential improvements

to be the core measurement process: the *Prepare for Measurement* activity and the *Perform Measurement* activity. These activities are included in the measurement process in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 and mainly address the concerns of the measurement user. The other two activities, the *Establish and Sustain Measurement Commitment* activity and the *Evaluate Measurement* activity, provide a foundation for the core measurement process and provide feedback. In addition, the ISO/IEC/IEEE 15939:2017 also defines a measurement information model which describes the link between the information needs and measurement concepts. Moreover, it also describes how the relevant measures are quantified to address the information needs as well as supporting decision making. Table 2 shows the technical tasks related to the four activities defined by the standard.

According to Enríquez et al. [42], an additional advantage of the ISO/IEC/IEEE 15939:2017 is that it provides a guideline for the definition of software measurement plans taking into account organizational aspects. However, the successful implementation of software measurement programs in the context of small and medium-sized software organizations (which have between 11 and 250 staff) is still a challenging task. One major recognized problem is that the ISO/IEC/IEEE 15939:2017 is difficult to read and understand, even for the users for whom it is intended.

Research by Chorás et al. [43] states that the definition of measurement programs is even less encouraging in small-sized software organizations, since most of the better measurement program models and standards are not suited to these types of organizations. Nevertheless, lightweight guidelines have been developed to adapt some standards for planning and establishing measurement processes in small-sized software organizations while still having limitations, such as budget, effort, and resources, in view. In this regard, a lightweight approach requires less time and effort, resulting in a more cost effective implementation of a software process standard. A large-sized software enterprise can follow a bottom-up approach to implement such standards as it can first identify the most critical problems and then work toward finding a solution for them. In the context of a small-sized software organization, a lightweight approach provides the ability to improve specific process areas by taking only the relevant parts of a standard that would improve the overall performance of small-sized software enterprises.

With this aim in mind, a lightweight implementation of the ISO/IEC/IEEE 15939:2017 core activities was conducted in order to define and implement a measurement program for the requirements elicitation process of a small-sized software organization called RagaSoft.

3.2 Context

RagaSoft is a small-sized software organization based in Oaxaca, Mexico, which currently has 30 employees. The organization produces software products in different application domains (e.g., administration, accountability, and payroll management). This paper provides details of a case study that was carried out at RagaSoft, using the IEEE Std. 830–1998 [22] to obtain the SRS and refine the use cases. It is worth mentioning that it was not our intention to change the organization's way of working, and we decided to allow the RagaSoft requirements engineers to continue using this standard for the case study. While we are conscious that

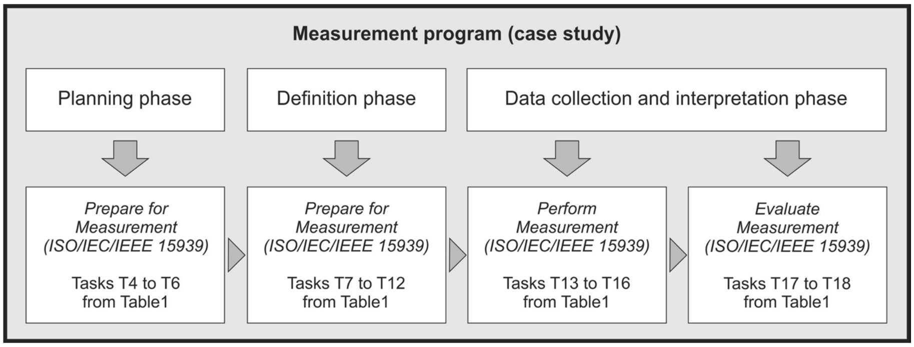

Measurement program (case study)
Planning phase
Definition phase
Data collection and interpretation phase
Prepare for Measurement (ISO/IEC/IEEE 15939)
Tasks T4 to T6 from Table1
Prepare for Measurement (ISO/IEC/IEEE 15939)
Tasks T7 to T12 from Table1
Perform Measurement (ISO/IEC/IEEE 15939)
Tasks T13 to T16 from Table1
Evaluate Measurement (ISO/IEC/IEEE 15939)
Tasks T17 to T18 from Table1

Fig. 1 The measurement program (case study) and the ISO/IEC/IEEE 15939:2017 activities

the introduction of other guidance from technical reports or standards (e.g., ISO/IEC TR 24766:2009, ISO/IEC/IEEE 29148:2018) could have significantly improved the requirements elicitation process, it would have required training leading to bias in the results of the case study. The handling and presentation of the data was authorized on the basis of a confidentiality agreement with RagaSoft.

3.3 Case study design

Researchers can evaluate their approaches in an objective and scientific way by performing an empirical study. According to Wohlin, Höst, and Henningsson [44], there are three types of empirical research for Software Engineering: surveys, case studies, and formal experiments. In this context, there are three strategies for developing a case study:

1. Compare the results obtained from a new proposal and the baseline.
2. Develop two projects in parallel (‘twin projects’), choosing one of them as a baseline.
3. Apply the new proposal to selected components and compare with the results from the components to which the new proposal had not been applied.

In order to provide relevant information about “how to do” the process, activities, and tasks defined by ISO/IEC/IEEE 15939:2017, we have followed the Wohlin et al. [44] guidelines for defining and conducting a case study related to the definition and execution of a measurement program in a specific small-sized software organization by adapting this standard. Moreover, because there is no evidence of similar works in literature for measuring and improving requirements elicitation activities with such a standard, we strongly believe, therefore, that we are providing initial data from positive results that must be supported with further case studies in other small organizations before providing more general guidelines which will increase the reproducibility of our study. Therefore, a positivist case study was designed as the experimental strategy within an industrial environment by choosing the first strategy. The list of preliminary requirements of this project was considered as the baseline to be compared with the list of preliminary requirements elicited after the measurement program. It is worth mentioning that the team who obtained this baseline was made up of three different people: a project manager and two requirements engineers[2] (i.e., the baseline team). Moreover, as a result of the measurement program, the elicitation activities were improved in comparison with those used when the baseline was obtained. The aim was to compare the results of using our proposal with the results of the baseline as well as avoiding bias while ensuring internal validity. A measurement team was comprised of three people playing the same roles as the baseline team. The participants, who had worked at RagaSoft for three years, had the same amount of experience and knowledge working with the requirements elicitation process and the project’s domain. In addition, two external experts also participated but were not part of the measurement team, and they were unaware that their work was part of this study. These “experts” were requirements engineers/research professors with more than 10 years’ experience in the field implementing measurement programs. The experts carried out a blind evaluation on both sets of preliminary requirements, not knowing anything about how they were obtained to evaluate their quality. RagaSoft used interviews and focus groups as the elicitation techniques for their projects to elicit a final preliminary requirements list. The project used for the measurement program was an informative web portal owned by a small organization to manage its internal information and the customer orientated services. With this aim in mind, the case study followed the three phases for the implementation of a measurement program: *planning*, *definition*, and *data collection and interpretation* (see Fig. 1).

In addition, Díaz, García, and Piattini [45] suggested that the definition of measures and research questions can add

[2] The requirements engineers fulfilled the roles of elicitors and analysts of the preliminary requirements, and they have skills and responsibilities to perform the requirements elicitation as well as writing skills for specifying the requirements.

Table 3 Definition of the measurement goal for identifying stakeholders

	Description
Goal	Identify the individual stakeholders or stakeholder classes (activity 1)
Purpose	Improve the stakeholder identification activity using the Volere template
Approach	The identified stakeholder classes must be relevant and necessary for the project
Perspective	The requirements engineer's viewpoint
Environment	In the context of a small project (six to twelve months and 50,000–100,000 LOC) that will be developed by a team of fewer than 10 members

Table 4 Definition of the measurement goal for integrating, refining, and organizing the collected information

	Description
Goal	Integrate, refine, and organize the collected information (activity 3)
Purpose	Analyze if all the obtained preliminary requirements by each stakeholder fulfill the quality criteria proposed by ISO/IEC/IEEE 29148:2018 [10] and Lauesen [21] (i.e., each requirement must be unambiguous, consistent, traceable (to its source or stakeholder), and understandable in order to be integrated into the final preliminary requirements list (to verify if it is complete))
Approach	The final list of preliminary requirements must be complete, unambiguous, consistent, traceable, and understandable
Perspective	The requirements engineer's viewpoint
Environment	In the context of a small project (six to twelve months and 50,000–100,000 LOC) that will be developed by a team of fewer than 10 members

formality to the definition of the basis of a measurement program, which enhances the chance of success and provides techniques which make the measurement process easier to define, use, reuse and understand within small-sized software organizations. With this aim in mind, two research questions were formulated as follows:

RQ1: is it possible to understand, control and improve the stakeholder identification using the proposed measurement program?

RQ2: is it possible to control and improve the quality of the final list of preliminary requirements using the proposed measurement program?

3.3.1 Planning and definition phases

The *Prepare for Measurement* activity (tasks T4–T12 from Table 1) of the ISO/IEC/IEEE 15939:2017 was implemented to cover the planning and definition phases as follows:

3.3.1.1 Define the measurement strategy (Task T4) According to the ISO/IEC/IEEE 15939:2017, a measurement strategy should be the objective of any software organization to obtain more accurately targeted and measured results from software development. The strategy must be aligned with an organization's business objectives to demonstrate the value of the measurement program to top management. Moreover, with the aim of simplifying this task for small-sized software organizations, Díaz et al. [45] suggested the use of a goal-driven measurement process to define a strategy that aligns measures and indicators with goals to show success in achieving such goals. In this way, the measures and indicators selected were used to show success in achieving the defined goals.

Therefore, the Goal-Question-Metric (GQM) paradigm [46, 47] was used to define, adjust, and evaluate a set of goals based on RagaSoft's objectives by using a measurement process. GQM considers three levels for formally defining a measurement program: (1) The *conceptual level* (Goal) that establishes the measurement goals for an object related to products, processes, or resources. These goals can be defined using a template for an object [48], (2) The *operational level* (Question) that generates a set of questions used to specify how the goals must be reached, and (3) The *quantitative level* (Metric) that defines the measures to quantitatively answer the question(s) established for the previous level [49]. GQM provides a hierarchical structure where the goals, questions and measures can have a many-to-many relationship. Moreover, due to the volatility of RE, improving the requirements elicitation process is a challenge, but quality assurance can be achieved by developing and applying adequate (valid) measures and measurements [50].

Despite its importance, stakeholder identification as well as the analysis of their needs and expectations is often poorly executed in software projects, which leads to uncontrollable and failed projects. Therefore, it is very likely that improving the way in which the industry carries out the requirements elicitation activities would lead to a higher success rate of software projects [51, 52]. With this aim in mind, Tables 3 and 4 show the definition of the measurement goals (the GQM conceptual level) for activity 1 (*identifying the* *individual stakeholders or stakeholder classes)* as well as for activity 3 (*integrating, refining, and organizing the collected information)*.

It is worth mentioning that the use of the Volere[3] template [53] was introduced in order to help the requirements engineers to identify the stakeholders (i.e., this template was used because it defines some stakeholder classes that may be present within the software development project). We believe that the advantages of this strategy were twofold for the context of a small-sized software organization. Firstly, it did not require an advanced level of theoretical knowledge, and secondly, it provided a good vision of preliminary requirements when data are being collected.

3.3.1.2 Describe the characteristics of the organization that are relevant to measurement (Task T5) According to the ISO/IEC/IEEE 15939:2017, this activity involves the characteristics that are relevant for selecting measures and interpreting the information products. In this context, such characteristics are the size of the software team—fewer than 10 members, size of projects—50,000–100,000 LOC, and time spent on such a project—six to twelve months. As previously stated, the measurement program was carried out with 3 employees from RagaSoft, using the IEEE Std. 830–1998 to obtain the SRS and refine the use cases. The list of preliminary requirements of a previous project was considered as the baseline to be compared with the list of preliminary requirements elicited after the measurement program, while the participating experts performed a blind evaluation on both sets of preliminary requirements. It is worth emphasizing once more that the measurement program focused on two activities related to the requirements elicitation process, activity 1: *identifying the individual stakeholders or stakeholder classes*, and activity 3: *integrating, refining, and organizing the collected information*. Therefore, the aim was to measure and improve the stakeholder identification activity and verify the quality of the final list of the elicited preliminary requirements.

3.3.1.3 Identify and prioritize the information needs (Task T6) The ISO/IEC/IEEE 15939:2017 states that the information needs are based on the organization's business objectives, the project objectives, identified risks and/or other items related to project decisions. In fact, Díaz et al. [45] stated that the aim when identifying information needs must be to set well-defined links between the expected results and the business context.

Therefore, considering the goals established by the measurement strategy, the information needs were identified for the two activities of the requirements elicitation process. For activity 1, the need for analyzing the stakeholder identification process to improve it and determine the necessary and relevant sources of requirements was defined. For activity 3, it would be necessary to analyze the processes for integrating, refining, and organizing the information to verify that the information is complete. Thus, two information needs were defined by following the ISO/IEC/IEEE 15939:2017 recommendations which states that if measurement is being undertaken for the first time within an organization, it is preferable to start on a small scale. Moreover, in order to ensure that a resulting process/product has quality, it was necessary to define indicators which describe the process/product by performing a measurement process [54]. In this regard, the *Measurement Information Model* defined by the ISO/IEC/IEEE 15939:2017 provides both principles and practices for developing, operating, and continuously improving a measurement program. Therefore, we tried to add formality to the preparation for the measurement by describing instantiations of the *Measurement Information Model* that addressed the defined information needs (see Tables 5 and 6).

3.3.1.4 Select and specify measures that satisfy the information needs (Task T7) A measure is a fundamental or unit-specific term, while a measure can literally be derived from one or more measures. In this regard, according to the ISO/IEC/IEEE 15939:2017, the same *Measurement Information Model* can be used to help with this task by establishing a clear link between the information needs and candidate measures.

Taking into account the defined goals, the *Measurement Construct Template* (defined by the standard) was used to link the variables that were measured to our information needs. Within the context of a small-sized software organization, these measurement constructs are useful in describing how the relevant activities and products of the requirements elicitation process are quantified and converted into indicators that provide a basis for decision making. The quality of the requirements elicitation process will influence the quality of the software because a minimal error at this stage will affect the remaining life cycle stages. Moreover, an effective specification of requirements will allow requirements engineers to provide better support when changing management activities, and thereby obtaining better results in testing, thus decreasing risk, improving the quality, and supporting the automatization. However, Díaz et al. [45] suggested the definition of a limited number of measures with the aim of reducing the overall complexity of the interpretation. Therefore, Table 7 shows the formal definition of eight measures proposed for measuring the activities 1 (M1–M3) and 3 (M4–M8) of requirements elicitation, respectively.

[3] The Volere template for stakeholder analysis is a spreadsheet that enables the requirements engineer to match types of knowledge with the roles in organization that provide the needed knowledge. According to Robertson and Robertson [53], the analysis spreadsheet is an effective way of recording and keeping track of all the stakeholders.

Table 5 Measurement construct for analyzing the stakeholder identification

Measurement construct	
Information need	Evaluate the stakeholder identification process in order to properly determine the necessary and relevant sources of requirements
Information category	Product quality
Measurable concept	1. Effectiveness
Relevant entities	1. Stakeholders 2. Preliminary requirements
Attributes	1. Stakeholder classes 2. Elicited preliminary requirements
Base measures	1. Number of stakeholder classes identified 2. Number of preliminary requirements proposed by each stakeholder 3. Total number of elicited preliminary requirements
Measurement method (by measure)	1. Count the number of stakeholder classes that participated in the elicitation process 2. Count the number of preliminary requirements proposed by each stakeholder 3. Count the number of total elicited preliminary requirements
Type of method (for all measures)	Objective
Scale (by measure)	1. Integers, from 0 to 14 2. Integers, from 0 to infinity[a] 3. Integers, from 0 to infinity
Type of scale (for all measures)	Ratio
Unit of measurement (by measure)	1. Identified stakeholder classes 2. Individual preliminary requirements 3. Total preliminary requirements
Derived measure	1. Stakeholder appropriateness (i.e., number of proposed preliminary requirements) per class 2. Individual appropriateness of each identified stakeholder
Measurement function	1. Divide the total number of preliminary requirements proposed by the stakeholder class that are in the final preliminary requirements list by the total number of elicited preliminary requirements: $P_s = \frac{c_s}{n}, where$ $\mathrm{P_s}$ is the stakeholder appropriateness, $\mathrm{c_s}$ is the number of preliminary requirements proposed by each stakeholder class, and n is the total number of elicited preliminary requirements
Indicator	1. Number of requirements proposed by each stakeholder per elicitation session 2. Average appropriateness of each stakeholder class
Analysis model	1. Identify a minimum of three stakeholder classes: client/sponsor, core team members (project manager, business analyst, requirements analyst, systems analyst, tester, technical writer, systems architect, systems designer), and external consultants (negotiator) [56] 2. Compute the average appropriateness of each stakeholder class
Decision criteria	1. If the minimum number of stakeholder classes are not identified, there is a risk that the preliminary requirements list will be incomplete 2. If a stakeholder did not propose at least one preliminary requirement during the elicitation sessions, it could indicate that he/she is not interested in the project or has no knowledge on the problem that needs solving

[a] According to McGarry et al. [55], a scale is an ordered set of values, continuous or discrete, or a set of categories to which an attribute is mapped. This scale defines the range of possible values that can be produced by executing the measurement method. For the case of numerical data, it is normal to consider from 0 to positive infinity

3.3.1.5 Define data collection, analysis, access, and reporting procedures (Task T8) Once the measures were defined, it was also necessary to define the procedures for collecting them. With the aim of creating clear and repeatable procedures, Annex F (informative) of the ISO/IEC/IEEE 15939:2017 states that this definition should normally describe who is responsible for measurement, the data source, the frequency of the data collection and the required tool to support this activity. Both requirements elicitation activities 1 and 3 can be divided into tasks and a template was designed for recording measurements. Moreover, a new activity was also proposed to analyze the collected data and provide proper data and timely feedback to the RagaSoft requirements engineers about the requirements elicitation process undertaken for the project (*Activity 4: Postmortem*). Therefore, Table 8 describes the procedures used to collect data at RagaSoft.

The previously defined procedures required the creation of templates for recording the measurement data, both for recording the stakeholders' information and analyzing the

Table 6 Measurement construct for analyzing the integration, refinement, and organization of information

Measurement construct	
Information need	Analyze the processes for integrating, refining, and organizing the information (elicited preliminary requirements) to verify that this information is complete and meets the ISO/IEC/IEEE 29148:2018 [10] and Lauesen [21] quality attributes
Information category	Product quality
Measurable concept	1. Quality of the elicited preliminary requirements 2. Completeness of the preliminary requirements list
Relevant entities	1. List of total preliminary requirements (e.g., for the evaluation of completeness) 2. Preliminary requirements (e.g., for the evaluation of ambiguity and consistency)
Attributes	1. Consistent preliminary requirements 2. Unambiguous preliminary requirements 3. Understandable preliminary requirements 4. Traceable preliminary requirements 5. Completeness of the preliminary requirements list
Base measures	1. Number of preliminary requirements in conflict 2. Number of interpretations for each preliminary requirement in the list 3. Number of preliminary requirements understood by all the stakeholders 4. Number of preliminary requirements that are associated with a stakeholder 5. Number of stakeholders who approved and signed the final version of the preliminary requirements list
Measurement method (by measure)	1. Count the number of preliminary requirements in conflict 2. Count the number of different interpretations of each preliminary requirement in the list 3. Count the number of preliminary requirements understood by all the stakeholders 4. Count the number of preliminary requirements that are associated with a stakeholder 5. Count the number of stakeholders who approved and signed the final version of the preliminary requirements list
Type of method (for all measures)	Objective
Scale (by measure)	1. Integers, from 0 to total number of preliminary requirements – 1[a] 2. Integers, from 0 to the total number of preliminary requirements 3. Integers, from 0 to the total number of preliminary requirements 4. Integers, from 0 to the total number of stakeholders 5. Integers, from 0 to the total number of stakeholders
Type of scale (for all measures)	Ratio
Unit of measurement (by measure)	1. Preliminary requirements in conflict 2. Understandability of preliminary requirements 3. Ambiguity of preliminary requirements 4. Traceability of preliminary requirements 5. Completeness of the preliminary requirements list
Derived measure	None
Measurement function	None
Indicator	1. Number of preliminary requirements in conflict when the information is integrated 2. Number of ambiguous preliminary requirements 3. Number of preliminary requirements understood by all stakeholders 4. Number of traceable preliminary requirements 5. Number of stakeholders who signed the final list of preliminary requirements
Analysis model	Some changes after the preliminary requirements review are expected. Therefore, a small number of changes are expected and can be accommodated. Additionally, all the preliminary requirements that fulfill the quality attributes proposed by ISO/IEC/IEEE 29148:2018 [10] and Lauesen [21] will be integrated into the final version of the preliminary requirements list
Decision criteria	1. If the number of preliminary requirements in conflict are equal to or greater than one, another elicitation session must be held with all stakeholders to solve conflicts among preliminary requirements 2. If the number of preliminary requirements understood are less than the total number of elicited preliminary requirements, another elicitation session must be held with all stakeholders to solve the problem 3. If the number of interpretations for each preliminary requirement is greater than one, it would be necessary to conduct another elicitation session to solve the problem 4. If the number of preliminary requirements that are associated with a stakeholder is less than the total number of elicited preliminary requirements, it would be necessary to conduct another elicitation session to solve the problem 5. If the number of stakeholders who approved and signed the final version of the preliminary requirements list is less than the total number of identified stakeholders, it would be necessary to conduct another elicitation session to solve the problem

[a] According to the ISO/IEC/IEEE 29148:2018 standard, a requirement is consistent if it is free of conflict with other requirements. Therefore, because there is at least one comparison between two preliminary requirements, at least one preliminary requirement from the list must be consistent (i.e., the one selected to begin the comparisons)

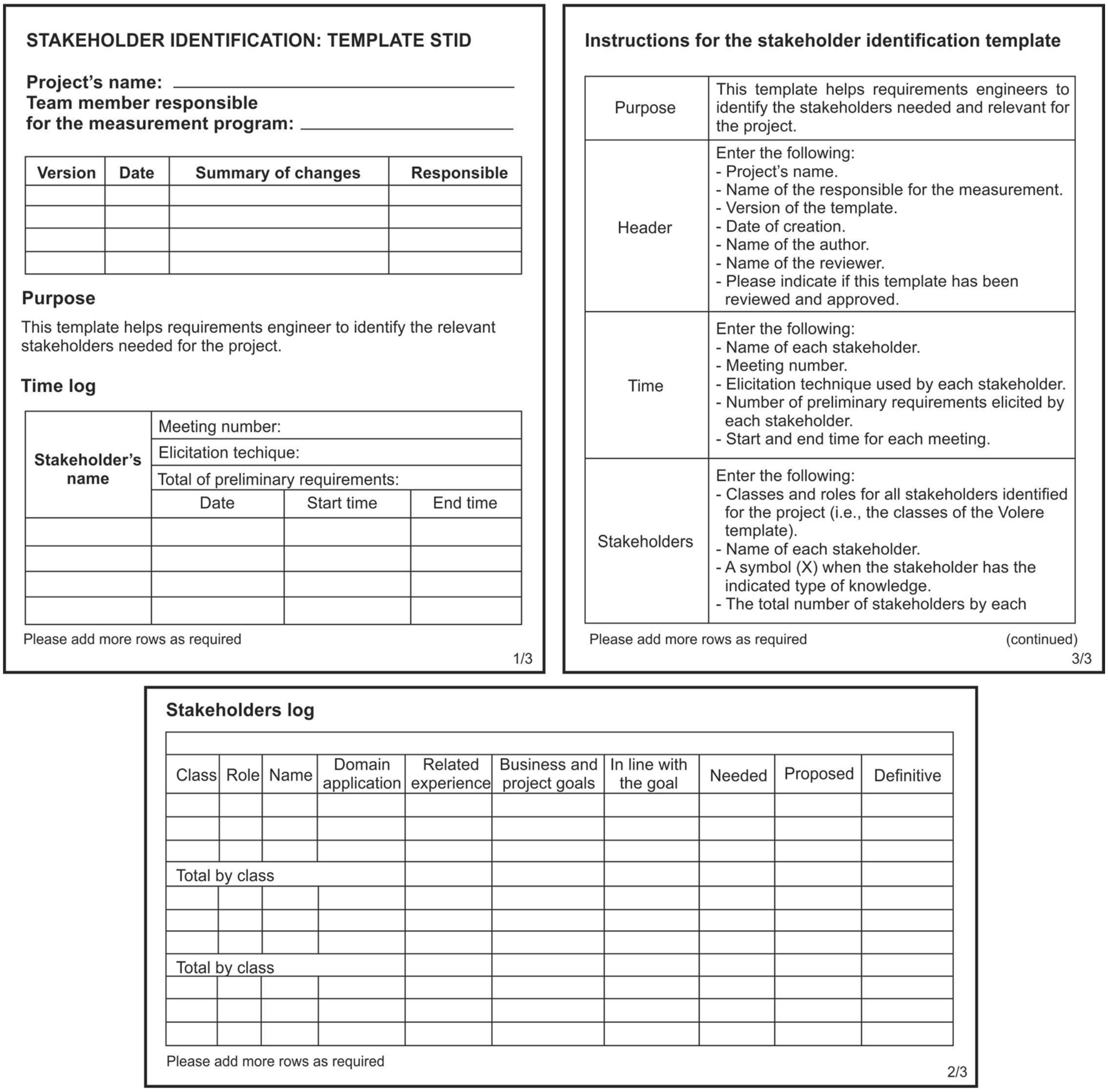
Table 7 Measures for the requirements elicitation activities and products

Measure	Definition	Unit of measurement	Range
M1. Number of stakeholder classes that participated in the elicitation process	Stakeholder classes[a] identified by the Volere template	Classes of stakeholders	1–14
M2. Number of preliminary requirements proposed by each stakeholder	Preliminary requirements proposed by each stakeholder identified according to the Volere template	Individual pre- requirements	0 to infinity
M3. Number of elicited pre- requirements	Total of preliminary requirements elicited in all the elicitation sessions	Total pre- requirements	0 to infinity
M4. Number of inconsistent pre- requirements	Preliminary requirements in conflict with others	Preliminary requirements in conflict	0 to (the total number of preliminary requirements—1)
M5. Number of ambiguous pre- requirements	Different interpretations of each stakeholder for each preliminary requirement in the list	Uncertain pre- requirements	0 to the total number of stakeholders
M6. Number of understandable pre- requirements	Preliminary requirements understood by all the stakeholders	Acceptable pre- requirements	0 to (the total number of preliminary requirements)
M7. Number of traceable pre- requirements	Preliminary requirements that are associated with a stakeholder	Attributable pre- requirements	0 to the total number of stakeholders
M8. Completeness of the list of preliminary requirements	All the stakeholders must have approved and signed the final version of the pre- requirements list	Stakeholder signatures on the final version of the list of preliminary requirements	0 to the total number of stakeholders

[a] The stakeholder classes define the roles that the stakeholders can perform regarding the Volere template for stakeholder analysis. According to Robertson and Robertson [53], there are 14 classes: interfacing technology, maintenance operator, normal operator, operational support, client, functional beneficiary, interfacing technology, internal consultant, sponsor, customer, interfacing technology, external consultants, negative stakeholders, and core team members

Table 8 Procedures for collecting data from the requirements elicitation activities (1, 3, and postmortem)

Activity 1: Identifying stakeholders	
Purpose	Provide a guideline for performing stakeholder identification
Input criteria	Project's description Template proposed for stakeholder identification Volere template for stakeholder analysis
Task 1: Analyze the stakeholders involved in the project	Use the Volere template to identify all the individual stakeholder and stakeholder classes involved in the project
Output criteria	Template proposed for the stakeholder identification (completed with the stakeholder analysis information) List of stakeholders who participated in the elicitation process taking into account the Volere template (completed)
Activity 3: Integrating, refining, and organizing the information	
Purpose	Provide a guideline for performing the integration, refinement, and organization of the information
Input criteria	List of stakeholders for each Volere class who participated in the elicitation process List of preliminary requirements for each stakeholder Template proposed for controlling the versions of the preliminary requirements Chronometer (optional) for recording the length of a stakeholder meeting
Task 1: Integrate the information	Integrate the preliminary requirements elicited from each stakeholder into a general list
Task 2: Refine the information	Analyze the elicited information for each preliminary requirement to ensure that the preliminary requirements are unambiguous, consistent, traceable, and understandable Record and resolve the conflicts of each preliminary requirement
Task 3: Organize the information	Write each refined preliminary requirement in a list All stakeholders must sign the final version of the list of preliminary requirements in order to verify its completeness
Output criteria	List of preliminary requirements Template proposed for controlling the versions of the preliminary requirements (fulfilled)
Activity 4: Postmortem (added in the context of this study)	
Purpose	Provide a guideline for performing the postmortem
Input criteria	Project's description Template proposed for stakeholder identification (fulfilled) Template proposed for controlling the versions of the preliminary requirements (fulfilled) Volere templates for stakeholder analysis (fulfilled) List of stakeholders who participated in the elicitation process (complete) Number of requirements that do not meet the quality attributes for each stakeholder Final list of preliminary requirements (complete) Template proposed for controlling the versions of the preliminary requirements (fulfilled)
Task 1: Collect learned lessons to document effective practices	Analyze the list of identified stakeholders Analyze the preliminary requirements from each stakeholder versus the list of the elicited preliminary requirements Fulfill the template proposed for controlling the versions of the preliminary requirements
Output criteria	Total effort spent on each activity of the requirements elicitation process (stored) Number of stakeholders identified using the Volere template Number of requirements elicited in each elicitation session Lessons learned (stored)



STAKEHOLDER IDENTIFICATION: TEMPLATE STID

Project's name: ____________________

Team member responsible
for the measurement program: ____________________

Version	Date	Summary of changes	Responsible

Purpose

This template helps requirements engineer to identify the relevant stakeholders needed for the project.

Time log

Stakeholder's name	Meeting number:		
	Elicitation techique:		
	Total of preliminary requirements:		
	Date	Start time	End time

Please add more rows as required

1/3

Instructions for the stakeholder identification template

Purpose	This template helps requirements engineers to identify the stakeholders needed and relevant for the project.
Header	Enter the following: - Project's name. - Name of the responsible for the measurement. - Version of the template. - Date of creation. - Name of the author. - Name of the reviewer. - Please indicate if this template has been reviewed and approved.
Time	Enter the following: - Name of each stakeholder. - Meeting number. - Elicitation technique used by each stakeholder. - Number of preliminary requirements elicited by each stakeholder. - Start and end time for each meeting.
Stakeholders	Enter the following: - Classes and roles for all stakeholders identified for the project (i.e., the classes of the Volere template). - Name of each stakeholder. - A symbol (X) when the stakeholder has the indicated type of knowledge. - The total number of stakeholders by each

Please add more rows as required (continued)

3/3

Stakeholders log

Class	Role	Name	Domain application	Related experience	Business and project goals	In line with the goal	Needed	Proposed	Definitive
Total by class									
Total by class									

Please add more rows as required

2/3

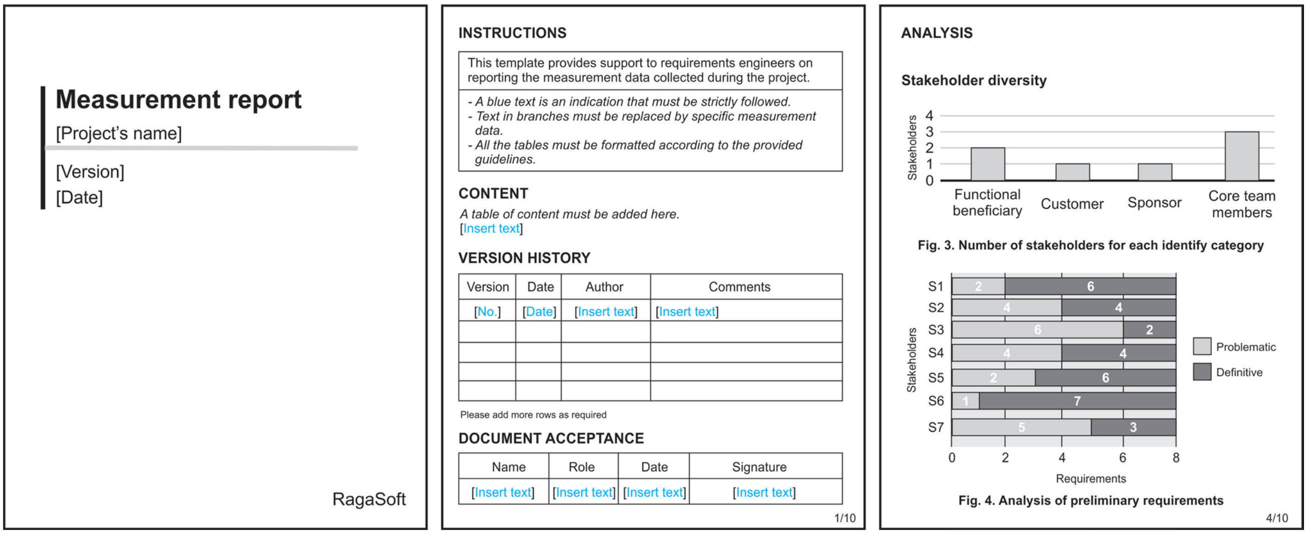
Fig. 2 Stakeholder identification template

information from the elicited preliminary requirements. These templates included a sheet of instructions so that they could be completed correctly (see Fig. 2).

In addition, to make the measurement program easier for the RagaSoft requirements engineers, Excel spreadsheets were designed for storing and processing the measurement data and recording the historical data. Therefore, all the data collected by using the defined templates were transcribed in the spreadsheets, which have the same format and include some additional fields to calculate the total values from each column. Moreover, the following analysis was proposed:

- In the context of *identifying the individual stakeholders or stakeholder classes*, it was necessary to record the stakeholders identified by each Volere class, the total number of stakeholders, and the total number of preliminary requirements proposed by each stakeholder.
- In the context of *controlling the quality of the preliminary requirements*, it was necessary to identify the preliminary requirements with some conflict or inconsistency, the number of preliminary requirements proposed by each stakeholder, the number of stakeholders who had a different interpretation of a preliminary requirement, the number of stakeholders who had clearly understood a preliminary requirement, and the number of stakeholders who had signed and approved the final version of the list of preliminary requirements.

Finally, once the measurement process was concluded, it was necessary to define how the collected data would be



Measurement report

[Project's name]

[Version]
[Date]

RagaSoft

INSTRUCTIONS

This template provides support to requirements engineers on reporting the measurement data collected during the project.

- *A blue text is an indication that must be strictly followed.*
- *Text in branches must be replaced by specific measurement data.*
- *All the tables must be formatted according to the provided guidelines.*

CONTENT

A table of content must be added here.
[Insert text]

VERSION HISTORY

Version	Date	Author	Comments
[No.]	[Date]	[Insert text]	[Insert text]

Please add more rows as required

DOCUMENT ACCEPTANCE

Name	Role	Date	Signature
[Insert text]	[Insert text]	[Insert text]	[Insert text]

1/10

ANALYSIS

Stakeholder diversity

Fig. 3. Number of stakeholders for each identify category

Fig. 4. Analysis of preliminary requirements

4/10

Fig. 3 Measurement report template

communicated to RagaSoft (e.g., establish a way to communicate the measurement data, the format of the final report, indicate the frequency, and the mechanisms for communication, as well as create the procedures for the distribution and availability of the data). A reporting mechanism was defined to present the measurement goals, the data related to each measure (e.g., frequency, relationships among measures, type of charts to better explain the measurement results), and the analysis technique. Thus, a measurement report (see Fig. 3) was created by the RagaSoft requirements engineers to communicate the measurement results by considering the following information:

- The frequency of the extracted data establishes the frequency rate at which the report should be created and stated.
- The reporting mechanisms indicate the format of the information that will be communicated.
- The data distribution establishes who should receive the measurement report and have access to the information collected by each measurement.
- The availability of the measures limits access for their reading, use, or modification.

3.3.1.6 Define criteria for evaluating the information items and the measurement process (Task T9) The ISO/IEC/IEEE 15939:2017 states that these criteria would allow requirements engineers to determine whether the required data have been collected and analyzed with sufficient quality to satisfy the information needs. In some cases, the criteria can be used for a quantitative evaluation, and in other situations, a qualitative evaluation may be appropriate.

Therefore, it was necessary to describe the aspects that must be analyzed to answer the measurement questions and then identify the existing relationships among the measurements collected. This procedure allowed the RagaSoft requirements engineers to identify potential problems in a timely manner and take proper corrective actions (i.e., if the collected measurements are significantly different from the original plans or if they are outside the established limits).

The following points provided an analysis for each of the previously defined measurement goals (see Table 4):

- *Identifying the individual stakeholders or stakeholder classes*: the requirements engineers analyzed which stakeholders would participate in the requirements elicitation process by considering the classes proposed by the Volere template [53]. Once the preliminary requirements were elicited, the stakeholders' appropriateness was analyzed to ensure that all their perspectives were considered. For example, considering activity 1: *identifying the individual stakeholders or stakeholder classes* related to Goal 1, Question 2: Were all the software perspectives obtained? and the Measure: "Number of preliminary requirements proposed by each stakeholder" (see Table 7), evaluates how many preliminary requirements, proposed by each stakeholder, were included into the final list of preliminary requirements. Consequently, the appropriateness of stakeholders could be determined. Therefore, the potential problem that was considered for this goal was the following:

  o Lack of appropriateness, if a stakeholder did not propose at least one preliminary requirement during the elicitation sessions. This could indicate that he/she is not involved, interested or suitable for the project.

- *Integrating, refining, and organizing the collected information*: the requirements engineers analyzed the elicited preliminary requirements to verify that the final list of preliminary requirements considered all the stakeholders' perspectives, needs, and interests. Regarding the integration activity, it was necessary to consider the different viewpoints from all stakeholder classes in relation to the number of preliminary requirements that had a particular problem (e.g., ambiguity, understandability, traceability, etc.). For example, taking into account activity 3: *integrating, refining, and organizing the collected information* related to Goal 2, Question 1: Do the elicited preliminary requirements fulfil the quality criteria? and the Measures (see Table 7): "Number of inconsistent preliminary requirements" and "Number of ambiguous preliminary requirements," if the number of ambiguous and inconsistent preliminary requirements is higher than the lower limit proposed by the range (i.e., 0), then the final list of preliminary requirements will contain fewer high-quality preliminary requirements. Moreover, the refinement task required the evaluation of the completeness of the final list of preliminary requirements to ensure that there was no extra or missing information (i.e., anything that was not specified). Thus, the potential problems that were considered for this goal were the following:

    o Lack of consistency, if there is any conflict among the preliminary requirements (i.e., there was no negotiation or conciliation among stakeholders).
    o Ambiguity, if any preliminary requirement was not understood in the same way by all the stakeholders, then this preliminary requirement is ambiguous. The higher the value of the measurement (i.e., the number of interpretations), the more ambiguity there will be for the preliminary requirement.
    o Difficulty in understanding the preliminary requirements, if the preliminary requirements have to be explained in detail to a stakeholder, then these were incorrectly written or require additional information in order to be understood. Therefore, the value of the measurement should be equal to the total number of elicited preliminary requirements to be sure that all the preliminary requirements have been understood by all the stakeholders.
    o Lack of traceability, if any preliminary requirement does not have a source, then it is untraceable.
    o Incomplete list of preliminary requirements, if the total number of stakeholders did not approve nor sign the final list of preliminary requirements (i.e., there are omissions in the preliminary requirements), then this list is incomplete.

In order to capture several performance aspects of the measures through their application, it was necessary to define some response variables. These response variables, which are related to RQ1 and RQ2, were derived from the measures proposed by activity A7 to measure the *identifying the individual stakeholders or stakeholder classes* and the *integrating, refining, and organizing the collected information* activities. The resulting response variables were as follows:

- Corresponding to **RQ1**:

    - Diversity of stakeholders. How many stakeholders have been identified by each Volere class?
    - Stakeholder appropriateness. What is the percentage of elicited preliminary requirements on the final list that are proposed by each identified stakeholder?

- Corresponding to **RQ2**:

    - Consistency of a preliminary requirement. How many preliminary requirements are in conflict?
    - Ambiguity of the preliminary requirements. How many interpretations of a preliminary requirement does each stakeholder have?
    - Traceability of a preliminary requirement. How many stakeholders have proposed preliminary requirements?
    - Understandability of the preliminary requirements. How many preliminary requirements have been understood by the stakeholders with a single explanation?
    - Completeness of the list of preliminary requirements. How many stakeholders have signed and approved the final version of the list of preliminary requirements?

With the aim of answering these questions, we examined the project documentation, observed participants at work, and evaluated each elicited preliminary requirement that was included in the final list of preliminary requirements as part of the defined measurement program.

3.3.1.7 Identify and plan for the enabling systems or services to be used (Task T10) According to the ISO/IEC/IEEE 15939:2017, enabling systems or services for measurement commonly include the tools and services used to collect, store, analyze, and report information. However, it has been reported that small-sized software organizations cannot allocate significant resources (e.g., human, technological, economic) to measurement programs due to the limited resources, which explains why only a three people were involved in this measurement program (a project manager and two requirements engineers). In addition, in order to avoid some of the major problems, challenges, and issues of implementing measurement programs in small-sized software organizations (e.g., low measurement maturity, poor software measurement knowledge, lack of experienced professionals), user-friendly templates were designed to collect the measure data. Moreover, Excel spreadsheets were designed to store, analyze, and report information.

3.3.1.8 Review, approve, and provide resources for measurement tasks (Task T11) All the previous measurement activities and tasks were reviewed and approved by participants (i.e., one project manager and two requirements engineers), the two external experts, and two members of the RagaSoft top management. Moreover, responsibilities and evaluation criteria of the measurement process were also reviewed, modified, and finally approved after eight iterations by top management. The availability of the participating employees and the schedule programmed for collecting data were also agreed with top management to achieve the commitment to measurement and, consequently, implement the measurement program.

3.3.1.9 Acquire and deploy supporting technologies (Task T12) For the final activity of the *Prepare the Measurement* activity, the ISO/IEC/IEEE 15939:2017 suggests evaluating and selecting, as appropriate, supporting technology such as automated tools or training courses. Moreover, the types of automated tools that are often needed include graphical presentation tools, data analysis tools, databases and/or tools for collecting data. However, as previously stated, there is a lack of resources, specialized staff, and automated tools for implementing measurement programs in small-sized software organizations [38]. Therefore, Excel and the designed templates were the only supporting tools used during the implementation of the measurement program.

3.3.2 Data collection and interpretation phase

The *Perform Measurement* activity (tasks T13–T18 from Table 1) of the ISO/IEC/IEEE 15939:2017 was implemented to cover the data collection and interpretation phase as follows:

3.3.2.1 Integrate procedures for data generation, collection, analysis, and reporting into the relevant processes (Task T13) The ISO/IEC/IEEE 15939:2017 states that for the first task of the *Perform Measurement* activity, all the created procedures should be integrated into the current organizational and project processes. This integration involved a slight modification of current processes to accommodate data generation and collection tasks. This modification involved the use of the templates designed to collect and present the measurement data (see Figs. 2 and 3).

Therefore, the project manager and the two requirements engineers from the measurement team, formed in Sect. 3.3, integrated all the defined procedures into their activities related to the project selected in Sect. 3.2. Moreover, the requirements elicitation activities were modified slightly so that the project manager and/or the requirements engineers had to fill out the templates and transcribe the information into the Excel spreadsheets when they recorded each preliminary requirement on the list. Furthermore, the two external experts supervised the correct execution of these activities, while the measurement team also used the Volere template [53] to perform the stakeholder identification activity. Additionally, the elicited preliminary requirements were recorded in the meeting minutes for each session.

3.3.2.2 Collect, store, and verify data (Task T14) The data were collected in three ways: a review of archival data (the preliminary requirements list used as the baseline), observation of participants, and measures (including the use of templates to collect the measurements). During the review of the baseline, the two expert researchers checked if it was properly carried out and evaluated the quality attributes of each elicited requirement. For the observation of participants, two requirements elicitation sessions were organized to obtain firsthand information of the problems relating to the requirements elicitation activities. The procedures for data collection were previously explained in Sect. 3.3.1.5.

The measurement program was contextualized to the requirements elicitation activities as follows:

- *Identifying the individual stakeholders or stakeholder classes*. All the possible sources of requirements (stakeholders) that could provide information were identified by using the Volere template [53] for stakeholder analysis taking into account the knowledge domain of the project to be developed. The participating requirements engineers identified the following stakeholders: software maintainer (maintenance operator class), operational technical user (normal operator class), chief executive (client class), business data modeler (internal consultant class), negotiator, security specialist (external consultant class) and the core team member's class. Additionally, the "stakeholders" section of the template for stakeholder identification was completed with the information required to indicate the stakeholders' type of knowledge. It is worth mentioning that the core team members class was not taken into account for the requirements elicitation activities as these stakeholders did not propose any preliminary requirement. This was because their participation was limited to only eliciting the preliminary requirements from the rest of the stakeholder classes. The core team members participated in the analysis, specification, verification and validation, and management of requirements. This activity lasted about eight hours/person.

- *Obtaining the list of preliminary requirements from each stakeholder by using the elicitation techniques*. The requirements engineers used two techniques for requirements elicitation: interviews and focus groups. It is important to mention that these techniques were used when the baseline project was developed, and to avoid any bias in this study, the participants used the same techniques. Two elicitation sessions were conducted to elicit the preliminary requirements. For the first elicitation session, interviews were only conducted with the operational technical user and the chief executive which lasted 10 h/person. From these interviews, 48 preliminary requirements were elicited (42 from the operational technical user and 6 from the chief executive). During the second elicitation session using a focus group, all the stakeholders participated in the focus group, including the two requirements engineers and the project manager to accommodate the diverse interests of the participants. Thus, 8 preliminary requirements were elicited, 30 preliminary requirements were modified, and 3 preliminary requirements were deleted with this second elicitation session lasting 17 h/person. The effort spent on interviews and the focus group was recorded into the template for stakeholder identification.
- *Integrating, refining, and organizing the collected information*. The preliminary requirements elicited by each stakeholder were integrated into a final list of preliminary requirements. Each elicited preliminary requirement was identified by the acronym PR*n*-*v*, where *n* is the number of the preliminary requirement and *v* is its version (e.g., PR1-1 refers to the first version of the preliminary requirement number 1). The template proposed for controlling the versions of the preliminary requirements was used to record the source of each preliminary requirement as well as all changes made to it. This activity lasted about 35 h/person.

3.3.2.3 Analyze data and develop information items (Task T15)

The analysis was applied to the quantitative data (i.e., measures) obtained in the requirements elicitation activities. After the measurement program was concluded, the expert researchers carried out a completely blind evaluation (i.e., the experts did not have any details on how these preliminary requirements were obtained) of the list of preliminary requirements obtained (baseline and this study) to validate their quality. Table 9 summarizes the data collected and analyzed during the requirements elicitation activities for the measurement program. The "Collected value" column indicates the values recorded for each measurement within the program and, for a better interpretation, its definition, and the range of allowed values are also shown. It is worth mentioning that the data collected in this study were quantitative and then validated by the expert researchers. The validation of the list of preliminary requirements involved reviewing their ambiguity, consistency, traceability, understandability, and completeness. After this validation, all the collected measurements and data were recorded in the defined database for processing and reporting results. Furthermore, additional data related to the performance of the measurement team and the baseline project participants were analyzed from the program.

3.3.2.4 Record results and inform the measurement users (Task T16)

Table 10 shows the results obtained (based on data shown in Table 9). The following findings were observed for each elicitation activity:

- *Identifying the individual stakeholders or stakeholder classes*: during the development of this activity, the measurement team:
  - o Identified five Volere stakeholder classes [53]: maintenance operator, normal operator, client, internal consultant, and external consultant.
  - o Regarding stakeholder appropriateness, the most appropriate stakeholders were the client, who proposed 29 preliminary requirements ($P_s = 29/53 = 0.54$), and the normal operator, who proposed 18 preliminary requirements ($P_s = 18/53 = 0.34$), it means 47 preliminary requirements of a total of 53 preliminary requirements were elicited by these two stakeholders.

- *Integrating, refining, and organizing the collected information*: the collected measurements provided the following evidence:
  - o The task related to collecting information about the inconsistency of the preliminary requirements which allowed requirements engineers to find 7 inconsistent preliminary requirements (13% of 53 preliminary requirements elicited). Nevertheless, conducting a focus group session resolved the inconsistencies among stakeholders.
  - o The task related to collecting information about the ambiguity of the preliminary requirements which allowed requirements engineers to identify 8 preliminary requirements with different interpreta-

Table 9 Data collected by implementing the measurement program in RagaSoft

Measure	Definition	Unit of measurement	Collected value	Range
M1. Number of stakeholder classes that participated in the elicitation process	Stakeholder classes identified by the Volere template that participated in the elicitation process	Stakeholder classes	5 classes: 1 Maintenance operator (software maintainer) 1 Normal operator (operational technical user) 1 Client (chief executive) 1 Internal consultant (business data modeler) 2 External consultants (security specialist)	1–14
M2. Number of preliminary requirements proposed by each stakeholder	Preliminary requirements proposed by each stakeholder identified according to the Volere template	Individual preliminary requirements	Software maintainer—proposed 1 preliminary requirement Operational technical user—proposed 18 preliminary requirements Chief executive—proposed 29 preliminary requirements Business data modeler—proposed 3 preliminary requirements Security specialist—proposed 2 preliminary requirements	0 to infinity
M3. Number of elicited preliminary requirements	Total of preliminary requirements elicited in all elicitation sessions	Total preliminary requirements	53 preliminary requirements	0 to infinity
M4. Number of inconsistent preliminary requirements	Preliminary requirements in conflict with others	Preliminary requirements in conflict	7 preliminary requirements	0 to (Total number of preliminary requirements – 1)
M5. Number of ambiguous preliminary requirements	Different interpretations of each stakeholder for each preliminary requirement (PR) in the list	Uncertain preliminary requirements	PR3-2, PR5-2, PR12-3, PR20-2, PR31-2, PR38-2, PR45-2, PR47-3	1 to total number of stakeholders
M6. Number of understandable preliminary requirements	Preliminary requirements understood by all the stakeholders with a single explanation	Acceptable preliminary requirements	48 preliminary requirements	0 to total number of preliminary requirements
M7. Number of traceable preliminary requirements	Preliminary requirements that are associated to a stakeholder	Attributable preliminary requirements	53 preliminary requirements	1 to total number of stakeholders
M8. Completeness of the list of preliminary requirements	All the stakeholders must have approved and signed the final version of the preliminary requirements list	Stakeholder signatures on the final version of the list of preliminary requirements	6 stakeholders	0 to the total number of stakeholders

Table 10 Findings from the measurement program

Baseline project		Measurement program	
Diversity of stakeholders	Appropriateness of stakeholders per class	Diversity of stakeholders	Appropriateness of stakeholders per class
Client	0.39	Maintenance operator	0.019
		Normal operator	0.340
		Client	0.547
		Internal consultant	0.056
		External consultant	0.038
Coverage of each quality attribute (%)			
Consistent (70%), Unambiguous (0%)*, Understandable (45%), Traceable (80%), Completeness** (100%)		Consistent (87%), Unambiguous (85%), Understandable (91%), Traceable (100%), Completeness** (100%)	
31 preliminary requirements were elicited		53 preliminary requirements were elicited	

*This attribute was evaluated considering that only one stakeholder participated in the baseline project

**This attribute was evaluated considering the full list of preliminary requirements

tions (15% of 53 preliminary requirements elicited). Therefore, it was necessary to review them with the corresponding stakeholders to avoid these ambiguities.

- The task related to collecting information about the understandability of the preliminary requirements which allowed requirements engineers to find that among the stakeholders, only 5 preliminary requirements were not understood (9% of 53 preliminary requirements elicited).
- All the elicited preliminary requirements were traceable. Each identified stakeholder proposed at least one preliminary requirement and, consequently, the traceability of the preliminary requirements was positively validated (100% of 53 preliminary requirements elicited).
- Finally, all the stakeholders approved the final list of preliminary requirements, demonstrating its completeness (100% of 6 stakeholders identified).

3.3.2.5 Evaluate information products and the measurement process (Task T17) According to the ISO/IEC/IEEE 15939:2017, an information product is one or more indicators and their associated interpretations that address an information need. In this regard, Annex E (informative) provides example criteria for evaluating the performance of the measurement process such as timeliness, efficiency, or process compliance.

With the aim of presenting relevant information on efficiency and process compliance, Table 9 summarizes information in two main rows for each participating group. The first one lists the stakeholders and their appropriateness, while the second one specifies the quality attributes that were evaluated from the list of preliminary requirements elicited from each project. The appropriateness value for the baseline project was 0.39 because from the 31 preliminary requirements provided by the only stakeholder, a total of 19 preliminary requirements were in the final list. This contrasts with the measurement team where all 53 proposed preliminary requirements appeared in the final list after applying the measures. Thus, the differences (see last row of Table 10) between the preliminary requirements obtained by the baseline project and those obtained by the new project showed an increase in the quality of the elicited preliminary requirements as they better fulfilled the stakeholders' expectations and needs. Despite these results, it is important to mention that as a consequence of the training and implementation of the measurement program, more effort was required for the measurement team (120 h/person) than the participants of the baseline project to perform the requirements elicitation activities. The measurement team achieved a better coverage of the quality attributes for the elicited preliminary requirements, a better diversity of the identified stakeholders and a higher number of elicited appropriate preliminary requirements than the participants of the baseline project. Summarizing, the obtained results allowed us to positively respond the two research questions previously formulated.

The measurement program provided the RagaSoft requirements engineers with information to better control the two requirements elicitation activities (activity 1: *identifying the individual stakeholders or stakeholder classes*, and activity 3: *integrating, refining, and organizing the collected information*) and a solid basis for timely decision making to positively impact on the quality of the elicited preliminary requirements. However, the implementation of this measurement program had its disadvantages, such as effort spent on training, planning, developing, processing, and analyzing data, use of inappropriate measurements, interference in processes, risks of misinterpretation, among other factors. Because the defined measurements did not negatively affect the requirements elicitation activities (as these were easily collected), the findings obtained can allow us to carefully evaluate their effectiveness in similar contexts, for example other small-sized software organizations, smalls teams and similar project domains.

3.3.2.6 Identify potential improvements (Task T18) The main improvement actions identified in this study are related to the measurement process and are summarized as follows:

- An important aspect to note is that the eight proposed measures have to be appropriately used by the requirements engineers taking into account a solid knowledge of human factors and requirements engineering activities. Therefore, the effort spent on training could be reduced providing more accurate definitions of each measure, integrating examples on how to properly interpret each measure, or improving the templates with more accurate instructions for collecting the measurement data.
- Special attention must be paid to clearly present the results, measures, and improvements achieved with the measurement program with the measurement team as well as the rest of the staff in order to achieve proper levels of motivation leading to the continuous improvement of the requirements elicitation activities.
- An additional improvement action is related to introduce a less invasive approach to collect, analyze, and interpret the measurement data in order to considerably reduce the effort required by the measurement program.

3.4 Threats to validity

A statistical test was not used for the results achieved in this study as it was only focused on the analysis of comparing the results obtained from a new proposal (requirements elicitation activities modified as a consequence of the measurement program) and the baseline. However, there are threats that may affect the validity of the presented results, such as the following:

- Internal validity. Internal validity issues deal primarily with the causal issues of our results. Therefore, none of the participants knew beforehand about the preparation of this study to avoid modifying their traditional practices and thus preventing a bias occurring in our measurements. There was, as a result, no internal motivation to influence the results. However, we cannot be sure that the members of the measurement team were enthusiastic about the use of templates and measures, whether this group had more smart members than the baseline project participants or that the use of the Volere template could have helped the measurement team to identify the relevant stakeholders more easily.
- Construct validity. Construct validity issues arise when there are errors in measurement. Quantitative data were collected by using templates during the requirements elicitation activities to ensure that the results were the same regardless of the person who was analyzing the data. It is important to note that the collected data only reflected the measurement team's compliance during the requirements elicitation activities (by using the measurement program and the Volere template), and we cannot assume that the same results could be obtained in subsequent stages of software development (e.g., design and coding). Finally, it is possible that the Volere template is not necessary for projects smaller than the one from this case study.
- External validity. External validity refers to how generalizable the results of the study are beyond the sample that is actually studied. This means that there is a possibility that some problems emerged given the characteristics of the study: organization size (i.e., the size of the software staff is 11 to 50 people, size of projects is 50,000 to 100,000 LOC, and the time spent on the projects is six to twelve months), size and complexity of the project and the project's application domain.

4 Conclusion and further research

This study has presented the experiences of using the ISO/IEC/IEEE 15939:2017 for defining and implementing a measurement program for two of the three activities that comprise the requirements elicitation process. This measurement program focused on the context of a small-sized software organization. The elicited preliminary requirements must be reviewed and approved by all the identified stakeholders (e.g., to validate the completeness of the preliminary requirements list) and the quality of these preliminary requirements must be verified to ensure that they are consistent, unambiguous, traceable, and understandable. Therefore, by determining preliminary quality requirements, any related problems and other issues will be avoided for the subsequent stages of the software development process. This will avoid the costs incurred for fixing these defects later and the possibility of the project failing. Therefore, with the aim of understanding the behavior of the requirements elicitation process, it was necessary to propose eight measures for interpreting, controlling, and improving the elicitation activities as part of the defined measurement program. Moreover, these measures can help requirements engineers to improve the quality of the elicited preliminary requirements. Considering the results from this study, the following conclusions can be drawn:

- The definition and implementation of this measurement program can serve as basis for the design of other programs which aim to follow the ISO/IEC/IEEE 15939:2017 [41] to measure and improve the requirements elicitation process in the context of a small-sized software organization.
- The incorporation of the Volere template [53] into the measurement team's activities was the correct decision as it helped requirements engineers to improve the activity related to the identification of stakeholders.
- Similarly, regarding *Identifying the individual stakeholders or stakeholder classes'* activity, the Volere template was used by the measurement team to perform the stakeholder analysis. This approach led to more stakeholders being identified compared to identifying the unique stakeholder by the participants in the baseline project. This is because the use of the template gave the measurement team a better indication of who might be participating in the elicitation sessions.
- The measurement program allowed requirements engineers to detect potential problems with the preliminary requirements (i.e., 33 inconsistent, 3 ambiguous, and 2 non-understandable requirements were identified) and take proper corrective actions as soon as possible.
- In order to evaluate the derived measurement called "Stakeholder appropriateness per class," it was necessary to evaluate the project's knowledge domain in order to obtain a correct interpretation of the measurement (e.g., some stakeholders' appropriateness was recorded at less than 10% because their knowledge was only required for very specific cases).
- Unlike the baseline project, the measurement team used two elicitation techniques under our supervision. Therefore, when the second elicitation technique was applied (i.e., focus group), it was possible to eliminate some indicators related to the poor quality of the preliminary requirements (e.g., ambiguity, inconsistency). Consequently, there was no inconvenience on the part of the stakeholders for signing and approving the final list of preliminary requirements, thereby ensuring its completeness.
- In general terms, the measurement program contributes to the systematic and effective development and management of the requirements elicitation process by providing an appropriate schema for *identifying the individual stakeholders or stakeholder classes*, as well as for *integrating, refining, and organizing the elicited preliminary requirements*.

Therefore, within the context of this study, the collected data allowed us to argue that our measurement program helped the RagaSoft requirements engineers to understand, control and improve the requirements elicitation activities which resulted in improved elicited preliminary requirements. Nevertheless, it is necessary to implement more measurement programs (e.g., in other small-sized software organizations, with projects of different sizes and complexity) to improve the eight measures proposed in this paper. Moreover, it is worth mentioning that this study was conducted within only one organization which elicited the preliminary requirements for a single project. Therefore, it could be argued that the results obtained can be replicated for many small-sized software organizations and similar projects. As previously stated, the measurement program needs to be applied to more organizations and projects with the aim of analyzing and improving the interpretation of each measurement. It is logical to assume that more specific measures could be proposed to improve their adaptation to each one of the elicitation techniques and the second activity of the elicitation process (i.e., *obtaining the list of preliminary requirements from each stakeholder by using the elicitation techniques*) which was not included in our research as some measures for this activity already exist. Furthermore, important decisions were made to achieve a productive adaptation of the ISO/IEC/IEEE 15939:2017, among which are the following:

- Take into account the recommendations by Díaz et al. [45] for the definition of measurement programs in small-sized software organizations considering the specific needs, maturity, and limitations of RagaSoft. In addition, the experience of the researchers who participated in the definition and execution of measurement programs in small and medium-sized software organizations was also considered to provide full-time support to the measurement team.
- Start the measurement program as long as RagaSoft top management give their full commitment, including allocating human resources (i.e., measuring team), in order to carry out the presented case study.
- Define only two measurement needs for RagaSoft that were prioritized together with top management to make informed decisions on the most important problems in two of the three requirements elicitation activities with the objective of partially improving such a process.
- Create a small set of measures in order to reduce the overall complexity of measuring the requirements elicitation activities within RagaSoft.
- Design accessible and easy-to-use instruments to collect the required information throughout the measurement program, paying attention to avoid excessive invasion of workspaces.
- Being part of the measurement team during practically the entire study with the aim of facilitating the adaptation of the standard and the correct execution of the activities.

- Finally, it was decided that the training on the use of procedures, information collection and analysis instruments, filling out forms, and the standard itself would be given by the same researchers who carried out the adaptation of the standard and wrote this article.
- Unfortunately, it was not considered to evaluate the learning curve during the participant training and, as a consequence, objective data on the duration of learning of each proposed activity (i.e., continuous or interrupted task) or the learning/forgetting related to individuals (i.e., individual(s) that learn) were not collected. However, it can be stated that the participants' type of learning, in this case, was cognitive. Therefore, by not having evaluated the learning curve, it was not possible to determine the participants' performance during the execution of the measurement process, the time dedicated exclusively to learning the process during the training, as well as the time in which the process is carried out correctly once the measurement equipment is no longer supported by the researchers. It is important to mention that with the intention of providing more conclusive information, it is proposed using and evaluating this learning curve in further case studies. Finally, we provide the following suggestions for future work:
- Conduct an additional study on the measurement process interfaced to the management and engineering processes defined by the international standard ISO/IEC 29110—systems and software engineering—lifecycle profiles for Very Small Entities (VSEs).
- Measure the level of satisfaction of the requirements engineers when implementing our measurement program.
- An interesting future research project would be to develop a computational tool to automatize the process for incorporating the eight proposed measures into a measurement program focused on the requirements elicitation activities. As previously mentioned, expediting the data collection, information processing and analysis along with the analysis and interpretation of the measurements could considerably reduce the effort required by the measurement program.
- Another interesting field of further research is to create alternative measures to analyze other RE processes, more specifically the analysis, specification, verification and validation, and management of requirements.
- The above-mentioned recommendation is also an example of further research. Further measurement programs could be implemented to create a historical database of lessons learned for small-sized software organizations which provide solid support for their requirements engineers to predict results for similar projects. Additionally, this information could be valuable for evaluating the effectiveness and scalability of the proposed measures in large projects.

Funding None.

Data availability statement The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Leyva M, Hadar I, Aviv I (2019) A requirements engineering methodology for knowledge management solutions: Integrating technical and social aspects. Requirements Eng 24(4):503–521. https://doi.org/10.1007/s00766-018-0298-x
2. Kotonya G, Sommerville I (1998) Requirements engineering: processes and techniques. John Wiley and Sons, Chichester
3. Berenbach B, Paulish DJ, Kazmeier J, Rudorfer A (2009) Software & systems requirements engineering: in practice. McGraw-Hill Inc, New York, NY
4. Ali N, Cai H, Hamou-Lhadj A, Hassine J (2019) Exploiting parts-of-speech for effective automated requirements traceability. Inf Softw Technol 106:126–141. https://doi.org/10.1016/j.infsof.2018.09.009
5. Femmer H, Hauptmann B, Eder S, Moser D (2016) Quality assurance of requirements artifacts in practice: a case study and a process proposal. In: Abrahamsson P, Jedlitschka A, Duc AN, Felderer M, Amasaki S, Mikkonen T (eds) Product-Focused Software Process Improvement. Springer, Cham, pp 506–516. https://doi.org/10.1007/978-3-319-49094-6_36
6. Medeiros J, Vasconcelos A, Silva C, Goulão M (2018) Quality of software requirements specification in agile projects: a cross-case analysis of six companies. J Syst Softw 142:171–194. https://doi.org/10.1016/j.jss.2018.04.064
7. Bourque P, Fairle RE (2014) Guide to the software engineering body of knowledge (SWEBOK®). IEEE Computer Society Press, Los Alamitos
8. Sandhu RK, Weistroffer HR (2018) A review of fundamental tasks in requirements elicitation. In: Wrycza S, Maślankowski J (eds) Information Systems: Research, Development, Applications, Education. Springer, Cham, pp 31–44
9. Dar H, Lali MI, Ashraf H, Ramzan M, Amjad T, Shahzad B (2018) A systematic study on software requirements elicitation techniques and its challenges in mobile application development. IEEE Access 6:63859–63867. https://doi.org/10.1109/ACCESS.2018.2874981
10. International Organization for Standardization (2018). ISO/IEC/IEEE 29148:2018 - Systems and software engineering - Life cycle processes -Requirements engineering, Geneva, Switzerland
11. Chen F, Power N, Collins J J, Ishikawa F (2019) Contemporary requirements challenges and issues: an empirical study in 11 organizations. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19). ACM, pp 1592–1599. https://doi.org/10.1145/3297280.3297435

12. Ahmad S, Asmai SA (2016) Measuring software requirements quality following negotiation through empirical study. Int J Appl Eng Res 11(6):4190–4196
13. Palmer J D, Evans R P. (1994). Software risk management: requirements-based risk metrics, In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Humans, Information and Technology. IEEE, pp 836–841. https://doi.org/10.1109/ICSMC.1994.399939
14. Dutoit AH, Bruegge B (1998) Communication metrics for software development. IEEE Trans Software Eng 24(8):615–628. https://doi.org/10.1109/32.707697
15. Kaiya H, Saeki M (2006) Using domain ontology as domain knowledge for requirements elicitation. In: 14th IEEE International Requirements Engineering Conference. IEEE, pp. 189–198. https://doi.org/10.1109/RE.2006.72
16. Barragáns-Martínez AB, Pazos-Arias JJ, Fernández-Vilas A, García-Duque J, López-Nores M, Díaz-Redondo RP, Blanco-Fernández Y (2008) On the interplay between inconsistency and incompleteness in multi-perspective requirements specifications. Inf Softw Technol 50(4):296–321. https://doi.org/10.1016/j.infsof.2007.02.001
17. Nakatani T, Tsumaki T, Tsuda M, Inoki M, Hori S, Katamine K (2011) Requirements maturation analysis by accessibility and stability. In: 18th IEEE Asia-Pacific Software Engineering Conference. IEEE, pp 357–364. https://doi.org/10.1109/APSEC.2011.56
18. Zapata S, Torres E, Sevilla G, Aballay L, Reus M (2012) Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios, In: Proceedings of the XXXVIII Conferencia Latinoamericana en Informatica (CLEI). IEEE, pp 1–7. https://doi.org/10.1109/CLEI.2012.6427200
19. Obana M, Hanakawa N (2014) Process evaluation based on meeting quality of requirement analysis phase in software development projects. J Softw Eng Appl 7(10):828–843. https://doi.org/10.4236/jsea.2014.710075
20. Byun J, Rhew S, Hwang M, Sugumara V, Park S, Park S (2014) Metrics for measuring the consistencies of requirements with objectives and constraints. Requirements Eng 19(1):89–104. https://doi.org/10.1007/s00766-013-0180-9
21. Lauesen S (2002) Software requirements, styles and techniques. Pearson Education, Harlow
22. Institute of Electrical and Electronics Engineers (1998). IEEE Std. 830–1998 - Guide for developing system requirements specifications, Piscataway, NJ
23. Gotel, O. & Finkelstein, C. W. (1994). An analysis of the requirements traceability problem, In: proceedings of the first international conference on requirements engineering. IEEE, pp 94–101. https://doi.org/10.1109/ICRE.1994.292398
24. Kneuper R (2018) Software and software process measurement software processes and life cycle models. Springer, Berlin, pp 261–285
25. Tahir T, Rasool G, Gencel C (2016) A systematic literature review on software measurement programs. Inf Softw Technol 73:101–121. https://doi.org/10.1016/j.infsof.2016.01.014
26. Nordin A, Zaidi NHA, Mazlan NA (2017) Measuring software requirements specification. J Telecommun Electron Comput Eng 9(3–5):123–128
27. International Organization for Standardization (2001). ISO/IEC 14598 - Information Technology - Software Product Evaluation, Geneva, Switzerland
28. International Organization for Standardization (2004). ISO/IEC 9126 - Software Engineering - Product Quality, Geneva, Switzerland
29 International Organization for Standardization Standard (2007). ISO/IEC 15939:2007 - Systems and Software Engineering - Measurement Process, Geneva, Switzerland
30. CMMI Product Team (2010). Capability Maturity Model Integration (CMMI) for Development, Version 1.3, Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University
31. International Organization for Standardization (2014). ISO/IEC 25021:2014 - Systems and Software Engineering. Requirements and Evaluation of Systems Quality and Software Security (SQuaRE) - Quality Score Elements, Geneva, Switzerland
32. Palyagar B (2004) Measuring and influencing requirements engineering process quality in organizations. In: 9th Australian Workshop on Requirements Engineering (AWRE '04), Adelaide, Australia, pp. 95–109.
33. Niazi M (2005) An instrument for measuring the maturity of requirements engineering process. In: Bomarius F, Komi-Sirviö S (eds) Product-Focused Software Process Improvement. Springer, Berlin, pp 574–585
34. Niazi M, Cox K, Verner J (2008) A measurement framework for assessing the maturity of requirements engineering process. Softw Qual J 16:213–235. https://doi.org/10.1007/s11219-007-9033-4
35. Génova G, Fuentes JM, Llorens J, Hurtado O, Moreno V (2013) A framework to measure and improve the quality of textual requirements. Requirements Eng 18(1):25–41. https://doi.org/10.1007/s00766-011-0134-z
36. Bhatti SN, Usman M, Jadi AA (2015) Validation to the requirements elicitation framework via metrics. ACM SIGSOFT Softw Eng Notes 40(5):1–7. https://doi.org/10.1145/2815021.2815031
37. Pinto N, Tortosa N, Geat B C, Ibáñez L, Bollati V (2018) Quality evaluation of agile processes: Measurement of requirements management using AQF v2. In: 11th International Conference on the Quality of Information and Communications Technology (QUATIC). IEEE, pp 15–20. https://doi.org/10.1109/QUATIC.2018.00013
38. Tahir T, Rasool G, Noman M (2018) A systematic mapping study on software measurement programs in SMEs. e-Informatica Softw Eng J 12(1):133–165. https://doi.org/10.5277/e-Inf180106
39. Laporte CY, Alexandre S, Renault A (2008) Developing international standards for very small enterprises. Computer 41(3):98–101. https://doi.org/10.1109/MC.2008.86
40. Laporte C Y, Renault A, Alexandre S (2008b) The Application of International Software Engineering Standards in Very Small Enterprises. Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies (42–70). IGI Global
41. International Organization for Standardization Standard (2017) ISO/IEC/IEEE 15939:2017 - Systems and software engineering - measurement process, Geneva, Switzerland
42. Enríquez JG, Sánchez-Begines JM, Domínguez-Mayo FJ, García-García JA, Escalona MJ (2019) An approach to characterize and evaluate the quality of product lifecycle management software systems. Comput Stand Interfaces 61:77–88. https://doi.org/10.1016/j.csi.2018.05.003
43. Chorás M, Springer T, Kozik R, López L, Martínez-Fernández S, Ram P, Rodríguez P, Franch X (2020) Measuring and improving agile processes in a small-size software development company. IEEE Access 8:78452–78466. https://doi.org/10.1109/ACCESS.2020.2990117
44. Wohlin C, Höst M, Henningsson K (2003) Empirical research methods in software engineering. In: Conradi R, Wang AI (eds) Empirical Methods and Studies in Software Engineering. Springer, Berlin, pp 7–23
45. Díaz M, García F, Piattini M (2005) Defining, performing and maintaining software measurement programs: State of art, In: VI Simposio Internacional de Sistemas de Información e Ingeniería

de Software en la Sociedad del Conocimiento, Cartagena de Indias, Colombia, pp 13–23

46. Basili VR (1993) Applying the goal/question/metric paradigm in the experience factory, software quality assurance and measurement: a worldwide perspective. International Thomson Publishing, London
47. Basili VR, Weiss DM (1984) A methodology for collecting valid software engineering data. IEEE Trans Software Eng 10(6):728–738. https://doi.org/10.1109/TSE.1984.5010301
48. Basili V R, Heidrich J, Lindvall M, Munch J, Regardie M, Trendowicz A (2007) GQM^+ strategies -- aligning business strategies with software measurement, In: First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007). IEEE, pp 488–490. https://doi.org/10.1109/ESEM.2007.66
49. Basili VR, Caldiera VRBG, Rombach HD (1994) Goal question metric paradigm. Encycl Softw Eng 1:528–532
50. Méndez Fernández D, Wagner S, Kalinowski M, Felderer M, Mafra P, Vetrò A, Conte T, Christiansson MT, Greer D, Lassenius C, Männistö T, Nayabi M, Oivo M, Penzenstadler B, Pfahl D, Prikladnicki R, Ruhe G, Schekelmann A, Sen S, Spinola R, Tuzcu A, de la Vara JL, Wieringa R (2017) Naming the pain in requirements engineering. Empir Softw Eng 22(5):2298–2338. https://doi.org/10.1007/s10664-016-9451-7
51. Pacheco C, Garcia I, Reyes M (2018) Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. IET Software 12(4):365–378. https://doi.org/10.1049/iet-sen.2017.0144
52. Wagner S, Méndez Fernández D, Felderer M, Vetrò A, Kalinowski M, Wieringa R, Pfahl D, Conte T, Christiansson MT, Greer D, Lassenius C, Männistö T, Nayebi M, Oivo M, Penzenstadler B, Prikladnicki R, Ruhe G, Schekelmann A, Sen S, Spínola R, Tuzcu A, Vara JL, Winkler D (2019) Status quo in requirements engineering: A theory and a global family of surveys. ACM Trans Softw Eng Methodol 28(2):1–48. https://doi.org/10.1145/3306607
53. Robertson S, Robertson J (2006) Mastering the requirements process, 2nd edn. Addison-Wesley, New York
54. Meding W, Staron M. (2020) Making software measurement standards understandable. In: Proceedings of the international conference on evaluation and assessment in software engineering (EASE '20). ACM, pp 368–370. https://doi.org/10.1145/3383219.3383269
55. McGarry J, Card D, Jones C, Layman B, Clark E, Dean J, Hall F (2002) Practical software measurement: objective information for decision makers. Addison-Wesley, Boston
56. Pacheco C, Garcia I (2012) A systematic literature review of stakeholder identification methods in requirements elicitation. J Syst Softw 85(9):2171–2181. https://doi.org/10.1016/j.jss.2012.04.075

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.