Advances in Intelligent Systems and Computing 1416

Jezreel Mejia · Mirna Muñoz ·
Álvaro Rocha · Himer Avila-George ·
Gloria Monica Martínez-Aguilar *Editors*

New Perspectives in Software Engineering

Proceedings of the 10th International Conference on Software Process Improvement (CIMPS 2021)

Springer

Advances in Intelligent Systems and Computing

Volume 1416

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Indexed by DBLP, INSPEC, WTI Frankfurt eG, zbMATH, Japanese Science and Technology Agency (JST).

All books published in the series are submitted for consideration in Web of Science.

More information about this series at http://www.springer.com/series/11156

Jezreel Mejia · Mirna Muñoz ·
Álvaro Rocha · Himer Avila-George ·
Gloria Monica Martínez-Aguilar
Editors

New Perspectives in Software Engineering

Proceedings of the 10th International Conference on Software Process Improvement (CIMPS 2021)

Springer

Editors
Jezreel Mejia
Centro de Investigación
en Matemáticas A.C.
Unidad Zacatecas
Zacatecas, Mexico

Mirna Muñoz
Centro de Investigación
en Matemáticas A.C.
Unidad Zacatecas
Zacatecas, Mexico

Álvaro Rocha
Departamento de Engenharia Informática
Universidade de Coimbra
Coimbra, Portugal

Himer Avila-George
Centro Universitario de los Valles
University of Guadalajara
Jalisco, Mexico

Gloria Monica Martínez-Aguilar
Universidad Tecnológica de Torreón
Coahuila, Mexico

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-030-89908-0 ISBN 978-3-030-89909-7 (eBook)
https://doi.org/10.1007/978-3-030-89909-7

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Introduction

This book contains a selection of papers accepted for presentation and discussion at the 2021 International Conference on Software Process Improvement (CIMPS'21). This conference had the support of the CIMAT A.C. (Mathematics Research Center/Centro de Investigación en Matemáticas); CANIETI (Coahuila-Durango Office); and UTT (Universidad Technological de Torreón Coahuila, México). It took place at UTT as virtual venue, in Torreón, México, from 20 to 22 October 2021.

The International Conference on Software Process Improvement (CIMPS) is a global forum for researchers and practitioners that present and discuss the most recent innovations, trends, results, experiences and concerns in the several perspectives of Software Engineering with clear relationship but not limited to software processes, security in information and communication technology and big data field. One of its main aims is to strengthen the drive towards a holistic symbiosis among academy, society, industry, government and business community promoting the creation of networks by disseminating the results of recent research in order to aligning their needs. CIMPS'21 is built on the successes of CIMPS'12, CIMPS'13, CIMPS'14, which took place on Zacatecas, Zac; CIMPS'15 which took place on Mazatlán, Sinaloa; CIMPS'16 which took place on Aguascalientes, Aguascalientes, México; CIMPS'17 which took place again on Zacatecas, Zac, México; CIMPS'18 which took place on Guadalajara, Jalisco, México; CIMPS'19 which took place on León, Guanajuato, México; and the last edition CIMPS'20 which took place on Mazatlán, Sinaloa, México, as virtual venue.

The Programme Committee of CIMPS'21 was composed of a multidisciplinary group of experts and those who are intimately concerned with software engineering and information systems and technologies. They have had the responsibility for evaluating, in a 'blind review' process, and the papers received for each of the main themes proposed for the conference are organizational models, standards and methodologies, knowledge management, software systems, applications and tools, information and communication technologies and processes in non-software domains (mining, automotive, aerospace, business, health care, manufacturing, etc.) with a demonstrated relationship to software engineering challenges.

CIMPS'21 received contributions from several countries around the world. The articles accepted for presentation and discussion at the conference are published by Springer (this book), and extended versions of best selected papers will be published in relevant journals, including SCI/SSCI and Scopus indexed journals.

We acknowledge all those who contributed to the staging of CIMPS'21 (authors, committees and sponsors); their involvement and support are very much appreciated.

October 2021

Jezreel Mejia
Mirna Muñoz
Álvaro Rocha
Himer Avila-George
Gloria Monica Martinez-Aguilar

Organization

Conference

General Chairs

Jezreel Mejía	Mathematics Research Center; Research Unit Zacatecas, Mexico
Mirna Muñoz	Mathematics Research Center, Research Unit Zacatecas, Mexico

The general chairs and co-chair are researchers in computer science at the Research Center in Mathematics, Zacatecas, México. Their research field is software engineering, which focuses on process improvement, multi-model environment, project management, acquisition and outsourcing process, solicitation and supplier agreement development, agile methodologies, metrics, validation and verification and information technology security. They have published several technical papers on acquisition process improvement, project management, TSPi, CMMI, multi-model environment. They have been members of the team that have translated CMMI-DEV v1.2 and v1.3 to Spanish.

General Support

CIMPS General Support represents centres, organizations or networks. These members collaborate with different European, Latin America and North America Organizations. The following people have been members of the CIMPS conference since its foundation for the last 10 years.

Gonzalo Cuevas Agustín	Polytechnical University of Madrid, Spain
Jose A. Calvo-Manzano Villalón	Politechnical University of Madrid, Spain
Tomas San Feliu Gilabert	Politechnical University of Madrid, Spain
Álvaro Rocha	Universidade de Lisboa, Portugal
Yadira Quiñonez	Autonomous University of Sinaloa, Mexico

Gabriel A. García Mireles	University of Sonora, Mexico
Adriana Peña Pérez-Negrón	University of Guadalajara, Mexico
Héctor Orozco-Aguirre	Autonomous University of México, Mexico

Local Committee

CIMPS established a local committee from the Mathematics Research Centre; Research Unit Zacatecas, Mexico, CANIETI Office Coahuila-Durango and the Technological University of Torreon, México. The list below comprises the local committee members.

CIMAT UNIT ZACATECAS:

Isaac Rodríguez Maldonado	Support, Mexico
Israel Faustino Cruz	Support, Mexico
Einar Jhordany Serna	Support, Mexico
Edgar Bonilla Rivas	Support, Mexico
Elizabeth Villanueva Rosas	Support, Mexico
Carmen Lizarraga	Support, Autonomous University of Sinaloa, Mexico
Raquel Aguayo	Support, Autonomous University of Sinaloa, Mexico
Mónica del Carmen Olivarría González	Designer, Autonomous University of Sinaloa, Mexico

CANIETI:

Hugo Montoya Díaz	President of the CANIETI Office, Mexico

UTT:

Raúl Martínez Hernandez	Provost, UTT, Mexico
Jesus Gerardo Serrano Mejia	Secretary of Networking, UTT, Mexico
Rogelio Martinez Mojica	Director of Networking, UTT, Mexico
Norma Patricia Mota Garcia	Deputy Director of Research and Development, UTT, Mexico
Gloria Monica Martinez Aguilar	UTT, Mexico

Scientific Programme Committee

CIMPS established an international committee of selected well-known experts in software engineering who are willing to be mentioned in the programme and to review a set of papers each year. The list below comprises the scientific programme committee members.

Adriana Peña Pérez-Negrón	University of Guadalajara, Mexico
Alejandro Rodríguez González	Politechnical University of Madrid, Spain
Alejandra García Hernández	Autonomous University of Zacatecas, Mexico
Álvaro Rocha	Universidade de Lisboa, Portugal
Ángel M. García Pedrero	Politechnical University of Madrid, Spain
Antoni Lluis Mesquida Calafat	University of Islas Baleares, Spain
Antonio de Amescua Seco	University Carlos III of Madrid, Spain
Baltasar García Perez-Schofield	University of Vigo, Spain
Benjamín Ojeda Magaña	University of Guadalajara, Mexico
Carlos Abraham Carballo Monsivais	CIMAT Unit Zacatecas, Mexico
Carla Pacheco	Technological University of Mixteca, Oaxaca, Mexico
Carlos Alberto Fernández y Fernández	Technological University of Mixteca, Oaxaca, Mexico
Claudio Meneses Villegas	Catholic University of North, Chile
Edgar Alan Calvillo Moreno	Technological University of Aguascalientes, Mexico
Edgar Oswaldo Díaz	INEGI, Mexico
Eleazar Aguirre Anaya	National Politechnical Institute, Mexico
Fernando Moreira	University of Portucalense, Portugal
Francisco Jesus Rey Losada	University of Vigo, Spain
Gabriel A. García Mireles	University of Sonora, Mexico
Giner Alor Hernández	Technological University of Orizaba, Mexico
Gloria P. Gasca Hurtado	University of Medellin, Colombia
Gonzalo Cuevas Agustín	Politechnical University of Madrid, Spain
Gonzalo Luzardo	Higher Polytechnic School of Litoral, Ecuador
Gustavo Illescas	National University of Central Buenos Aires Province, Argentina
Héctor Cardona Reyes	CIMAT Unit Zacatecas, Mexico
Héctor Duran Limón	University of Guadalajara CUCEA, Mexico
Himer Ávila George	University of Guadalajara, Mexico
Hugo Arnoldo Mitre Hernández	CIMAT Unit Zacatecas, Mexico
Hugo O. Alejandrez-Sánchez	National Center for Research and Technological Development, CENIDET, Mexico
Iván García Pacheco	Technological University of Mixteca, Oaxaca, Mexico
Jezreel Mejía Miranda	CIMAT Unit Zacatecas, Mexico
Jorge Luis García Alcaraz	Autonomous University of Juárez City, Mexico

José Alberto Benítez Andrades	University of Lion, Spain
Jose A. Calvo-Manzano Villalón	Politechnical University of Madrid, Spain
José Antonio Cervantes Álvarez	University of Guadalajara, Mexico
José Antonio Orizaga Trejo	University of Guadalajara CUCEA, Mexico
José E. Guzmán-Mendoza	Polytechnic University of Aguascalientes, Mexico
José Guadalupe Arceo Olague	Autonomous University of Zacatecas, Mexico
José Luis Sánchez Cervantes	Technological University of Orizaba, Mexico
Juan Francisco Peraza Garzón	Autonomous University of Sinaloa, Mexico
Juan Manuel Toloza	National University of Central Buenos Aires Province, Argentina
Leopoldo Gómez Barba	University of Guadalajara, Mexico
Lisbeth Rodriguez Mazahua	Technological University of Orizaba, Mexico
Lohana Lema Moreta	University of the Holy Spirit, Ecuador
Luis Omar Colombo Mendoza	Technological University of Orizaba, Mexico
Luz Sussy Bayona Ore	Autonomous University of Peru
Magdalena Arcilla Cobián	National Distance Education University, Spain
Manuel Mora	Autonomous University of Aguascalientes, Mexico
Manuel Pérez Cota	University of Vigo, Spain
María de León Sigg	Autonomous University of Zacatecas, Mexico
María del Pilar Salas Zárate	Technological University of Orizaba, Mexico
Mario Andrés Paredes Valverde	University of Murcia, Spain
Mary Luz Sánchez-Gordón	Østfold University College, Norway
Miguel Ángel De la Torre Gómora	University of Guadalajara CUCEI, Mexico
Mirna Muñoz Mata	CIMAT Unit Zacatecas, Mexico
Omar S. Gómez	Higher Polytechnic School of Chimborazo, Ecuador
Patricia María Henríquez Coronel	University Eloy, Alfaro de Manabi, Ecuador
Perla Velasco-Elizondo	Autonomous University of Zacatecas, Mexico
Ramiro Goncalves	University Tras-os Montes, Portugal
Raúl Aguilar Vera	Autonomous University of Yucatán, Mexico
Ricardo Colomo Palacios	Østfold University College, Norway
Roberto Solis Robles	Autonomous University of Zacatecas, Mexico
Santiago Matalonga	University of the West, Scotland
Sergio Galván Cruz	Autonomous University of Aguascalientes, Mexico

Sodel Vázquez Reyes	Autonomous University of Zacatecas, Mexico
Sonia López Ruiz	University of Guadalajara, Mexico
Stewart Santos Arce	University of Guadalajara, Mexico
Sulema Torres Ramos	University of Guadalajara, Mexico
Tomas San Feliu Gilabert	Politechnical University of Madrid, Spain
Ulises Juárez Martínez	Technological University of Orizaba, Mexico
Vianca Vega	Catholic University of North Chile, Chile
Víctor Saquicela	University of Cuenca, Ecuador
Viviana Y. Rosales Morales	University of Veracruz, Mexico
Yadira Quiñonez	Autonomous University of Sinaloa, Mexico
Yasmin Hernández	INEEL, Mexico
Yilmaz Murat	Çankaya University, Turkey

Contents

Organizational Models, Standards and Methodologies

A Proposal of Metrics for Software Requirements Elicitation in the Context of a Small-Sized Software Enterprise

C. Pacheco[1], I. García[1](✉), J. A. Calvo-Manzano[2], and M. Reyes[1]

[1] Division de Estudios de Posgrado, Universidad Tecnologica de la Mixteca, Carretera a Acatlima, 69000 Oaxaca, Mexico
{leninca,ivan,karenina}@mixteco.utm.mx
[2] Escuela Tecnica Superior de Ingenieros Informaticos, Universidad Politecnica de Madrid, Boadilla del Monte, 28660 Madrid, Spain
joseantonio.calvomanzano@upm.es

Abstract. The requirements elicitation process aims to obtain a preliminary version of the requirements before their specification. However, due the nature of this process, it is very difficult to know if the elicited requirements have quality. Furthermore, there are some metrics that allow requirements engineers to measure some aspects of the elicitation process and its final product (i.e., a preliminary version of the requirements, also called pre-requirements), however there is no way of measuring and controlling the elicited pre-requirements. With the aim of understanding, evaluating, and improving the requirements elicitation process, the existing metrics for this process have been analyzed and, as a consequence, eight metrics using the GQM paradigm are proposed. Additionally, a preliminary validation of these metrics has been performed in the context of a small-sized software enterprise. The obtained results are also presented.

Keywords: Requirements engineering · Requirements elicitation process · Metrics

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]. The RE process is frequently described by the following stages: elicitation, analysis, specification, validation and verification, as well as management [2]. Therefore, the final product of the RE process is the Software Requirements Specification (SRS) which establishes the functional and non-functional requirements for the software that will be developed. Nevertheless, a list of pre-requirements must be previously obtained by performing the elicitation and analysis stages. It is worth noting that the requirements elicitation process is not a trivial task and it is one of the main sources of confusion and causes many projects to fail [3, 4]. Thus, the improvement of the software quality depends on the SRS quality and, consequently, the quality of the

J. Mejia et al. (Eds.): CIMPS 2021, AISC 1416, pp. 3–14, 2022.
https://doi.org/10.1007/978-3-030-89909-7_1

pre-requirements should improve. According to Bourque and Fairley [5] and Sandhu and Weistroffer [6], the requirements elicitation process is frequently defined by three main activities: (1) the *stakeholder identification* (i.e., those people who provide the information needed to identify the problem to solve, as well as their needs and constraints); (2) *obtaining the list of pre-requirements of each stakeholder* by using the elicitation techniques; and (3) the *integration, refinement, and organization of the collected information* in order to determine the functionalities and limitations of the desired software. However, because human participation is essential for the elicitation process and the large quantity of information that needs to be handled, some problems may arise. For example, the crucial stakeholders do not participate in the elicitation sessions; there are conflicts among stakeholders; the elicited requirements do not reflect the required quality attributes; the time allocated for the elicitation process is not enough; unskilled requirements engineers; the stakeholders are not convinced about the need of a new software; or the lack of good communication and collaboration among stakeholders [7, 8].

With the aim of obtaining better results from the requirements elicitation process, measurement activity may be needed in order to improve the control of this process and provide positive feedback to all stakeholders (e.g., programmers, project management, customers). According to Basili [9], the ideal mechanism for improving a process is through measuring it and its products by establishing measurable objectives supported by the proper models. It is important to bear in mind that a measurement process can promote the maturity of the requirements elicitation process in order to help stakeholders understand what they really want and what they really need, as well as to recognize the constraints and limitations of the software that will be developed [10]. Furthermore, the quality attributes of the pre-requirements (i.e., the indicators for the metrics) are important for determining the quality of the product that will be obtained (i.e., list of pre-requirements) as these attributes can be measured by inspection (using the metrics) while the reported metrics can be used to individually evaluate the pre-requirements quality as well as to generally evaluate the full list of pre-requirements. Moreover, the historical data (obtained by the prolonged use of the metrics) can be used for the early identification of potential problems, and to carry out timely corrective actions in order to provide guidance about the decisions that should be taken. Therefore, in this paper, metrics for measuring and controlling the requirements elicitation process are presented to enhance this process by improving the quality to the elicited pre-requirements. The rest of the paper is organized as follows: Sect. 2 examines other approaches which introduce some metrics for the requirements elicitation process. With this aim in mind, Sect. 3 briefly describes the application of the GQM paradigm in order to design specific metrics for understanding and controlling the requirements elicitation process. Section 4 presents the preliminary results obtained in the context of a small-sized software enterprise. Finally, Sect. 5 summarizes the main conclusions of this paper.

2 Related Work

Software metrics have been an important research topic in Software Engineering for many years. In this context, the measurement process is an important instrument for understanding, evaluating or predicting, managing, and improving software entities such as processes, products, and resources [11]. Therefore, the use of metrics has been highlighted by several authors providing the possible reasons and effects of using metrics that could also be applied to the requirements elicitation process. For example, Palmer and Evans [12] proposed the Advanced Integrated Requirements Engineering System (AIRES) in order to provide automated support for the identification of requirements which are at risk of being ambiguous, redundant, inconsistent, untraceable, or incomplete. To this end, AIRES introduced three types of metrics: counting, ratio of risk, and relative size.

Research by Dutoit and Bruegge [13] developed and refined some metrics for obtaining the communication degree among stakeholders. The proposed metrics enabled researchers to collect data on artifacts generated by groupware tools (e.g., e-mail, memos, messages) during requirements elicitation. This study highlighted the use of communication metrics by arguing that they can provide better insights into software processes and products compared with metrics based on code.

Moreover, Kaiya and Saeki [14] proposed a set of metrics to measure the degree of domain knowledge in software application development during the requirements elicitation process. This study used an ontology to represent the terms and semantic structure of the problem domain to solve. Thus, by using these definitions, some metrics were proposed in order to evaluate the requirements quality and to identify incompleteness, inconsistency, ambiguity, or errors in the structural characteristics of the ontology and the software requirements.

Similarly, Barragáns et al. [15] proposed a metric for determining the level of inconsistency among the stakeholders' viewpoints (i.e., level of inconsistency in requirements, conflict degree, and the uncertainty of the viewpoints). The designed metric determines the close relationship between disagreement and incompleteness by obtaining the level of inconsistency, meaning that the best model (i.e., model of underspecified states that represents each viewpoint) reflects the combined knowledge of all stakeholders in the elicitation activities.

Nakatani et al. [16] highlighted that the requirements volatility indicates the level of maturity of the requirements (i.e., software requirements do not change if they achieve 100% maturity during the elicitation process), and defined a method for evaluating the Requirements Maturation Efficiency (RME) by taking into account stability and accessibility. The RME is obtained by dividing the number of requirements that remain unchanged by the total number of elicited requirements, while the accessibility is calculated as the inverse of the maximum distance of communication between the requirements analyst and the requirements source.

Research by Hanakawa and Obana [17] introduced some metrics to evaluate the quality of the elicitation sessions, because these directly influence the quality of the obtained requirements. These metrics were classified in basic (start time and end time of a speaking, stakeholder who is speaking, and the start time and end time of a theme) and advanced (average of a system engineer's speaking time, average of a stakeholder's speaking time, the number of times of system engineers' speakings, the number of times of stakeholders' speakings, ratio of system engineers' speaking time to all discussion time, ratio of stakeholders' speaking time to all discussion time, ratio of the number of times of system engineers' speakings to total number of time of speakings, ratio of the number of times of stakeholders' speakings to total number of time of speakings, and the maximum number of times of stakeholders' speakings between a system engineer speaking and a system engineer speaking).

Moreover, Zapata et al. [18] performed a study related to distributed software development in order to evaluate the efficiency of an elicitation technique by using the metrics proposed by Lloyd, Rosson, and Arthur [19] with adjustments for the requirements elicitation process. The metric used, called SRS document quality, takes into the account the following features: aspects related to the document's organization; the percentage of requirements that are identified as an evolution of a basic requirement; average of requirements without defects of precision, ambiguity, consistency, etc. (i.e., defects attributable to deficiencies in the requirements elicitation process); and the percentage of supported requirements which indicates the completeness of the produced document.

Finally, research by Byun et al. [20] emphasized the importance of the consistency of requirements as they must be compatible with the proposed objectives and constraints of the software that will be developed. This study proposed six metrics (degree of objective contribution, degree of objective satisfaction, value obtainable for each requirement, degree of constraint conformance, degree of constraint impact, and cost demandable for each requirement) in order to evaluate the requirements considering the objectives and constraints of the software. Additionally, five consistency metrics were designed: a metric to measure the consistency of each requirement that is related to one objective and various constraints, a metric to measure the consistency of each requirement that is related to one objective and one constraint, a metric to measure the consistency of each requirement that is related to various objectives and various constraints, a metric to measure the consistency of each requirement that is related to various objectives and one constraint, and finally a metric to measure the consistency of each requirement using its value and cost.

In summary, all of the proposed metrics only focused on the second activity of the requirements elicitation process (i.e., *obtaining the list of pre-requirements of each stakeholder by using the elicitation techniques*). However, the other two activities (the *stakeholder identification*, and the *integration, refinement, and organization of the collected information*) are not covered by the previous studies. Therefore, it is necessary to identify metrics that help requirements engineers to measure the elicitation progress and the quality of the pre-elicited requirements.

3 Eight Metrics for Measuring the Requirements Elicitation Process

The Goal-Question-Metric (GQM) paradigm [9, 21] represents a systematic approach to define, adjust, and evaluate a set of objectives based on the objectives of a project or organization as well as by using a measurement process. GQM takes into account 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 by using a template for an object [22], 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 metrics to quantitatively answer the question(s) established for the previous level [23]. GQM provides a hierarchical structure where the goals, questions, and metrics can have a many-to-many relationship. Moreover, improving RE is a challenge due to its volatility, however, quality assurance can be achieved by developing and applying appropriate (valid) metrics and measurements [24]. With this aim in mind, Tables 1 and 2 show the definition of the measurement goals (the GQM conceptual level) for *stakeholder identification*, and the *integration, refinement, and organization of the collected information.*

Table 1. Definition of the measurement goal for stakeholder identification.

	Description
Goal	Identify the project stakeholders
Purpose	Improve the stakeholder identification activity using the Volere template [25]
Approach	The identified stakeholders 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 2. Definition of the measurement goal for integration, refinement, and organization of the collected information.

	Description
Goal	Integrate, refine, and organize the collected information
Purpose	Analyze if all the obtained pre-requirements by each stakeholder fulfil the quality criteria proposed by IEEE Std. 830-1998 [26], Lauesen [27], and ISO/IEC/IEEE 29148 [28] (i.e., the final list of pre-requirements is complete, each requirement has to be unambiguous, consistent, traceable (to its source or stakeholder) and understandable)
Approach	The final list of pre-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

It is worth mentioning that the use of the Volere template [25] was introduced in order to help the requirements engineer 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 this strategy provided two main advantages in the context of a small-sized software enterprise: it did not require an advanced level of theoretical knowledge and it provided a good overview of the pre-requirements when data was being collected. Tables 3 and 4 provide the basis for the formulation of questions and metrics for each defined goal. Taking into account these goals, the *Measurement Construct Template* defined by the Practical Software Measurement (PSM) approach [29] was used to link the variables that were measured against our information needs (https://drive.google.com/file/d/1mknDJSpKSQcNS-38R7NmBpAy7wgwYjQK/view?usp=sharing). Within the context of a small-sized software enterprise, these measurement constructs are useful for 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.

Table 3. Metrics proposed for stakeholder identification.

Goal	Question	Metrics
Identify the project stakeholders	How many classes of stakeholders were identified by the Volere template?	# of classes of stakeholders that participated in the elicitation process
	Were all the software perspectives obtained?	# of pre-requirements proposed by each stakeholder
		# of elicited pre-requirements

Table 4. Metrics proposed for integration, refinement, and organization of information.

Goal	Question	Metrics
Integrate, refine and organize the information	Do the elicited pre-requirements fulfil the quality criteria [26–28]?	# of inconsistent pre-requirements
		# of ambiguous pre-requirements
		# of understandable pre-requirements
		# of traceable pre-requirements
		Completeness of the list of pre-requirements

The quality of the requirements elicitation process will influence the quality of the software development process because a minimal error at this stage would affect the rest of the life cycle stages. Moreover, an effective requirements definition will allow requirements engineers to remain disciplined during the software development process, provide better support when changing management activities, and obtain better results in testing, thereby decreasing risk, improving the quality, and supporting automatization. Table 5 shows the formal definition of the eight metrics proposed for measuring the requirements elicitation activities 1 and 3, respectively.

Once the metrics were defined, it was also necessary to define the procedures for collecting them. In order to make the procedures clear and repeatable, the ISO/IEC/IEEE 15939:2017 standard [30] states that this definition should normally describe who is responsible for the measurement, the data source, the frequency of the data collection, and the required tool for supporting this activity. Both requirements elicitation activities 1 and 3 can be divided into tasks, with a template designed for recording measurements. Moreover, a new activity was also proposed for the requirements elicitation process in order to analyze the collected data and provide proper data as well as timely feedback to the requirements engineer about the requirements elicitation process carried out in the project (*Activity 4: Post-mortem*). The defined procedures required the creation of templates for recording the measurement data, such as the stakeholders' information and analyzing the information of the elicited pre-requirements. These templates included a sheet of instructions to avoid any errors being made. In addition, with the aim of making the measurement program easier for the requirements engineer of the small-sized software enterprise, Excel spreadsheets were designed for storing and processing the measurement data and recording the historical data.

Table 5. Definition of metrics for the requirements elicitation products.

Metric	Definition	Unit of measurement	Range
# of classes of stakeholders that participated in the elicitation process	Classes[a] of stakeholders identified by the Volere template	Classes of stakeholders	1–14
# of pre-requirements proposed by each stakeholder	Pre-requirements proposed by each stakeholder identified according to the Volere template	Individual pre-requirements	0 to infinity
# of elicited pre-requirements	Total of pre-requirements elicited in all the elicitation sessions	Total pre-requirements	0 to infinity
# of inconsistent pre-requirements	Pre-requirements in conflict with others	Pre-requirements in conflict	0 to (the total number of pre-requirements – 1)
# of ambiguous pre-requirements	Different interpretations of each stakeholder for each pre-requirement on the list	Uncertain pre-requirements	0 to the total number of stakeholders
# of understandable pre-requirements	Pre-requirements understood by all the stakeholders	Acceptable pre-requirements	0 to (the total number of pre-requirements)

(*continued*)

Table 5. (*continued*)

Metric	Definition	Unit of measurement	Range
# of traceable pre-requirements	Pre-requirements that are associated with a stakeholder	Attributable pre-requirements	0 to the total number of stakeholders
Completeness of the list of pre-requirements	All the stakeholders must have approved and signed the final version of the pre-requirements list	Final version of the list of pre-requirements	0 to the total number of stakeholders

[a]The classes of stakeholders define the roles that they can perform with regard to the Volere template for stakeholder analysis. According to Robertson and Robertson [25], 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.

Finally, in order to validate the feasibility of this approach, a preliminary validation in a small-sized Mexican software enterprise was conducted. The following section provides the details.

4 Preliminary Results

An early validation was conducted in a small-sized software enterprise with 20 employees which used Scrum to develop the software projects and the IEEE Std. 830-1998 to obtain the SRS as well as refine the use cases. Previous to the requirements specification, the enterprise used two elicitation techniques in their projects: interviews and focus groups to elicit a pre-requirements list. The project used for this case study was an informative web portal owned by a small corporation for managing all of its internal information and the services focused on customers. This case study used, as a baseline, a pre-requirements list produced by a team who worked in the requirements elicitation process without using the defined measurement program (the *control group*) and another team who used the measurement program to evaluate its work in the context of this case study (the *experimental group*). These groups were comprised of three people: a project manager and two requirements engineers. The experimental group members were selected randomly because they all had the same experience and knowledge on the requirements elicitation process and the project's domain. All of the participants had worked at the enterprise for three years.

It is important to mention that no member of the control group had communication with any member of the experimental group, as both groups were comprised of different members, and none of the members of the experimental group knew beforehand about the preparation for this exercise. The experimental group used the Volere template [25] to perform the stakeholder identification activity. Furthermore, the control group did not carry out stakeholder identification because their members considered the client as the only stakeholder in the requirements elicitation process. Additionally, the elicited pre-requirements were recorded in the meeting minutes for each session.

The data was collected in three ways: review of archival data (pre-requirements list used as baseline), observation of participants, and metrics (including the use of templates to collect the measurements). During the review of the baseline, an expert researcher checked if it was properly carried out and evaluated the quality attributes of each requirement elicited for the control group. The "expert" is a requirements engineer/professor with more than 10 years' experience, was not part of the experimental group, and was also unaware that his work was part of the validation. The analysis was applied to the quantitative data obtained in the requirements elicitation activities. The expert carried out a completely blind evaluation of the list of pre-requirements obtained by both the control and experimental groups to evaluate their quality, and did not know anything about how they were obtained. For the observation of participants, two requirements elicitation sessions were conducted to get firsthand information concerning the problems of the requirements elicitation activities. Table 6 summarizes the obtained results.

Table 6 has two main rows for each group: the first one lists the stakeholders and their efficiency (rows 3 to 7 in Table 6), while the second one specifies the quality attributes that were evaluated from the list of pre-requirements elicited by each project (rows 8 and 9 in Table 6). The appropriateness value was 0.45 for the baseline project, because from the 31 pre-requirements provided by the only stakeholder, a total of just 19 pre-requirements were selected for the final list. This is in contrast to the experimental group, where all 53 proposed pre-requisites appeared in the final list after the metrics were applied.

Table 6. Preliminary results from the validation.

Control group		Experimental group	
Diversity of stakeholders	Appropriateness of stakeholders per classes	Diversity of stakeholders	Appropriateness of stakeholders per classes
Client	0.45	Maintenance operator	0.019
		Normal operator	0.340
		Client	0.547
		Internal consultant	0.056
		External consultant	0.038
Coverage of each quality attribute (%)			
Unambiguous* (0%), Consistent (70%), Traceable (80%), Understandable (45%), Completeness** (100%)		Unambiguous (85%), Consistent (80%), Traceable (100%), Understandable (76%), Completeness** (100%)	
31 pre-requirements were elicited		53 pre-requirements were elicited	

*This attributed was evaluated taking into account that in the baseline project, only one stakeholder participated.
**This attributed was evaluated taking into account the full list of pre-requirements.

Thus, the differences (see last row of Table 6) between the pre-requirements obtained by the baseline project (control group) and those obtained by the new project (experimental group) show an increase in the quality of the elicited pre-requirements as they better fulfill the stakeholders' expectations and needs. Despite these results, it is important to mention that as a consequence of the training and implementation of the metrics, an additional month was required for the experimental group to perform the requirements elicitation activities than the control group. The experimental group achieved a better coverage of the quality attributes for the elicited pre-requirements, a better diversity of the identified stakeholders, and a higher number of elicited appropriate pre-requirements than the participants of the baseline project. It is clear that these results are promising, however, they are not useful for generalizing the benefits of the metrics. Nevertheless, two measurement programs are currently being conducted to support this approach by providing more quantitative evidence.

5 Conclusions

We have proposed eight metrics for the requirements elicitation process which have to be reviewed by all the stakeholders in order to achieve their acceptation (i.e., pre-requirements validation). The quality of these pre-requirements must be verified to ensure that they are complete, correct, consistent, unambiguous, verifiable, traceable, and modifiable (i.e., pre-requirements verification). Therefore, as soon the characteristics and constraints of the software are determined, problems and errors will be avoided for the remaining stages of the software process development. Otherwise, the cost of fixing these errors will increase and the project might fail. It is worth mentioning that the presented validation was conducted within only one enterprise who elicited the pre-requirements for only one project, therefore we cannot argue that the obtained results can be replicated for other enterprises and projects. As previously stated, our metrics need to be applied to more enterprises and projects with the aim of analyzing and improving the interpretation of each one. We strongly believe that more specific metrics can 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 pre-requirements of each stakeholder* by using the elicitation techniques), which was not included in our research because there are some related metrics for this part.

References

1. Tukur, M., Umar, S., Hassine, J.: Requirement engineering challenges: a systematic mapping study on the academic and the industrial perspective. Arab. J. Sci. Eng. **46**, 3723–3748 (2021)
2. Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.: Requirements engineering: a systematic mapping study in agile software development. J. Syst. Softw. **139**, 32–50 (2018)
3. Beewolf, R.: Project excellence or failure? Doing is the best kind of learning. IEEE Eng. Manage. Rev. **44**(2), 26–32 (2016)
4. Dar, H., Lali, M.I., Ashraf, H., Ramzan, M., Amjad, T., Shahzad, B.: A systematic study on software requirements elicitation techniques and its challenges in mobile application development. IEEE Access **6**, 63859–63867 (2018)

5. Bourque, P., Fairley, R.E.: Guide to the Software Engineering Body of Knowledge (SWEBOK®), Version 3.0. IEEE Computer Society Press, Los Alamitos (2014)
6. Sandhu, R.K., Weistroffer, H.R.: A review of fundamental tasks in requirements elicitation. In: Wrycza, S., Maślankowski, J. (eds.) SIGSAND/PLAIS 2018. LNBIP, vol. 333, pp. 31–44. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00060-8_3
7. Ghazi, P., Glinz, M.: Challenges of working with artifacts in requirements engineering and software engineering. Requirements Eng. **22**(3), 359–385 (2017). https://doi.org/10.1007/s00766-017-0272-z
8. Dar, H.S.: Reducing ambiguity in requirements elicitation via gamification. In: 28th International Requirements Engineering Conference (RE 20), pp. 440–444. IEEE Computer Society (2020)
9. Basili, V.R.: Applying the goal/question/metric paradigm in the experience factory. In: Software Quality Assurance and Measurement: A Worldwide Perspective, pp. 21–44. International Thomson, London (1993)
10. Ohashi, K., Katayama, A., Hasegawa, N., Kurihara, H., Yamamoto, R., Doerr, J., Magin, D.P.: Focusing requirements elicitation by using a UX measurement method. In: 26th International Requirements Engineering Conference (RE 18), pp. 347–357. IEEE Computer Society (2018)
11. Tahir, T., Rasool, G., Gencel, C.: A systematic literature review on software measurement programs. Inf. Softw. Technol. **73**, 101–121 (2016)
12. Palmer, J.D., Evans, R.P.: Software risk management: requirements-based risk metrics. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 836–841. IEEE Computer Society (1994)
13. Dutoit, A.H., Bruegge, B.: Communication metrics for software development. IEEE Trans. Softw. Eng. **24**(8), 615–628 (1998)
14. Kaiya, H., Saeki, M.: Using domain ontology as domain knowledge for requirements elicitation. In: 14th IEEE International Requirements Engineering Conference (RE'06), pp. 189–198. IEEE Computer Society (2006)
15. Barragáns, A.B., et al.: On the interplay between inconsistency and incompleteness in multi-perspective requirements specifications. Inf. Softw. Technol. **50**(4), 296–321 (2008)
16. Nakatani, T., Tsumaki, T., Tsuda, M., Inoki, M., Hori, S., Katamine, K. Requirements maturation analysis by accessibility and stability. In: 18th IEEE Asia-Pacific Software Engineering Conference (APSEC 18), pp. 357–364. IEEE Computer Society (2011)
17. Hanakawa, N., Obana, M.: A metrics for meeting quality on a software requirement acquisition phase. In: Dieste, O., Jedlitschka, A., Juristo, N. (eds.) PROFES 2012. LNCS, vol. 7343, pp. 260–274. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31063-8_20
18. Zapata, S., Torres, E., Sevilla, G., Abdalla, L., Reus, M.: Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios. In: XXXVIII Conferencia Latinoamericana en Informatica (CLEI 12), pp. 1–7. IEEE Computer Society (2012)
19. Lloyd, W.J., Rosson, M.B., Arthur, J.D.: Effectiveness of elicitation techniques in distributed requirements engineering. In: IEEE Joint International Conference on Requirements Engineering, pp. 311–318. IEEE Computer Society (2002)
20. Byun, J., Rhew, S., Hwang, M., Sugumara, V., Park, S., Park, S.: Metrics for measuring the consistencies of requirements with objectives and constraints. Requirements Eng. **19**(1), 89–104 (2014)
21. Basili, V.R., Weiss, D.M.: A methodology for collecting valid software engineering data. IEEE Trans. Softw. Eng. **10**, 728–738 (1984)
22. Basili, V.R., Heidrich, J., Lindvall, M., Munch, J., Regardie, M., Trendowicz, A.: GQM+strategies -- aligning business strategies with software measurement. In: First International Symposium on Empirical Software Engineering and Measurement, pp. 488–490. IEEE Computer Society (2007)