UNIVERSIDAD
NACIONAL
DE COLOMBIA

A Pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110

Antonio Fernando Vera Delgado

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2019

A Pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110

Antonio Fernando Vera Delgado

Tesis de investigación presentada como requisito parcial para optar al título de:

Magister en Ingeniería—Ingeniería de Sistemas

Director:

Ph.D. Carlos Mario Zapata Jaramillo

Línea de Investigación:

Ingeniería de Software

Grupo de Investigación:

Lenguajes Computacionales

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2019

I dedicate this thesis to my beloved son, my mother, my father and my brother. They whom keep that light inside me always bright.

Acknowledgments

I would like to express my deep gratitude to my research project supervisor, Professor Carlos Mario Zapata, for his patient support, constructive suggestions, and useful critiques of this research work. It was my privilege to receive the guidance of a person with such knowledge and professionalism. Thank you for everything.

I would also like to extend my thanks and sincere gratitude to the Departamento de Ciencias de la Computación y la Decisión, which gave me an educational scholarship, allowing me the opportunity to finish this thesis.

Finally, I wish to thank my parents and my brother for their support and encouragement throughout my study.

Resumen

Estándares tales como la norma ISO/IEC 29110 se implementan y practican en pequeñas organizaciones (*Very Small Entities*, abreviado VSE) para mejorar la calidad del proceso de software. El perfil básico establecido en la guía de administración e ingeniería de dicha norma incluye un conjunto de buenas prácticas para mejorar el proceso en términos de tiempo, eficiencia, presupuesto y documentación. Por otro lado, los esquemas preconceptuales son representaciones de conocimiento creadas para modelar un dominio en un entorno gráfico, fácil e intuitivo. La cantidad de VSEs que adoptan el perfil básico viene aumentando, ya que la mayoría de los proyectos que desarrollan las VSEs son adecuados para este modelo de proceso. En los procesos detallados en los estándares se describe "qué hacer", dejando el "cómo hacer" a discreción de la organización dispuesta a implementar dicho estándar. En consecuencia, los procesos detallados en la norma ISO/IEC 29110 carecen de la prescripción de un método en particular; por lo tanto, las VSEs enfrentan problemas al implementar y desarrollar las actividades del estándar. Este hecho puede llevar a una posible interpretación errónea de la norma en las VSEs, al no lograr la calidad esperada e impedirles lograr los objetivos del proyecto, especialmente en primeras implementaciones. En esta Tesis de Maestría se propone un método basado en la norma ISO/IEC 29110 para educir requisitos en pequeñas organizaciones mediante el uso de esquemas preconceptuales ejecutables. Además, se valida el método propuesto realizando un caso de estudio en un proyecto de investigación que financia una entidad gubernamental. Al desarrollar un método de esta índole, se pretende consolidar una alternativa para VSEs al implementar el perfil básico de la norma ISO/IEC 29110 en etapas iniciales.

Palabras clave: ISO/IEC 29110, perfil básico, esquema preconceptual, ingeniería de software, pequeñas y medianas empresas.

Abstract

Standards such as the ISO/IEC 29110 have been adopted and practiced by Very Small Entities (VSE) for improving quality of the software process. The basic profile established in the management and engineering guide of such standard includes a set of best practices for enhancing the process in terms of time, efficiency, budget, and documentation. On the other hand, pre-conceptual schemas are knowledge representations created for modeling a domain in a graphical, easy, and intuitive environment. The amount of VSEs adopting the basic profile has increased, since most of the projects developed by VSEs are suitable for this process model. What-to-do is described in detailed standard processes, leaving the how-to-do-it choice at discretion of the organization willing to implement such a standard. Consequently, the processes detailed in the ISO/IEC 29110 lack prescription of a particular method and VSEs face a compliance problem when implementing and developing the standard activities. Such a fact could lead to possible misunderstanding of the standard by VSEs, failing to achieve the expected quality and moving away from the project objectives, especially in first-time implementations. In this M.Sc. Thesis, we propose a method based on the ISO/IEC 29110 for eliciting requirements in very small entities by using executable pre-conceptual schemas. Besides, we validate the proposed method by conducting a case study in a research project financed by a governmental entity. By developing such a method, we intend to consolidate an alternative to VSEs when implementing the basic profile of the ISO/IEC 29110 in early stages.

Keywords: ISO/IEC 29110, basic profile, pre-conceptual schema, software engineering, very small entities.

Table of contents

Pág.

Figures list

Pág.

Table list

Pág.

Abbreviation and symbol list

Abbreviations

Abbreviation	Concept
VSE	Very Small Entity
PM	Project management
SI	Software implementation
TL	Technical Leader
WT	Work Team
CUS	Customer
AN	Analyst
DES	Designer
PR	Programmer
UML	Unified Modeling Language
SME	Small and medium-sized enterprises
OECD	Organization for Economic Co-operation and Development
CMMI	Capability Maturity Model Integration
EPF	Eclipse Process Framework
OWL	Web Ontology Language
CEO	Chief Executive Officer

Introduction

Software process standards have been developed by several organizations in order to meet current demands for quality in terms of process, product, and company. According to the ISO/IEC (2016), a standard is a “document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context.” Software process standards (such as the ISO/IEC 29110 standard “Lifecycle profiles for Very Small Entities”) have been implemented for VSEs in different countries. VSEs try to improve the software development process (which usually includes an upgrade in terms of time, budget, human resources, customer satisfaction, quality, and other subjects) by implementing the ISO/IEC 29110. The generic profile group defined in the management and engineering guide of the systems engineering part of the ISO/IEC 29110 contains a set of profiles with typical situational factors of occurrence applicable to some VSEs (ISO/IEC, 2011). According to Laporte, Séguin, Boas, and Buasung (2013), “a profile is a kind of matrix that identifies which elements should be taken from existing standards.” Each profile has a core containing a different set of processes, activities, tasks, roles, etc. The basic profile described in the generic profile group is intended to improve quality in software and hardware projects and it is suitable to any development approach based on the VSEs needs (ISO/IEC 2011).

On the other hand, pre-conceptual schemas are graphical knowledge representations aimed at modeling any domain accurately and unambiguously (Zapata, 2007a). The development of such schemas helps to transform natural language discourse into a controlled language, which can be used to obtain conceptual models such as UML class diagrams, entity relationship models, case use diagrams, etc. Besides, pre-conceptual schemas can be executed interactively in order to provide a roadmap to the user for navigating within the domain.

The basic profile has been used in case studies and conferences in contrast to the other profiles established in this standard (Laporte & O’Connor, 2016; Sánchez & O’Connor, 2016; Laporte, 2011; Díaz, de Jesús, Melendez & Dávila, 2016; García, Laporte, Arteaga & Bruggmann, 2015; Siddoo, Wongsai & Wetprasit, 2013; O’Connor, 2014; Bougaa, O’Connor, Bornhofen & Rivière, 2017; Rodríguez & Buitrago, 2015). Such usage is based on the fact some projects have a size suitable for the lightweight process model (Project Management—PM—and Software Implementation—SI—processes) included in such a profile. Besides, the basic profile is suitable for only one application at a time, developed by a single project team with no situational factors or special risk (ISO/IEC 2011).

Even though standards provide all kinds of benefits, they are unrelated to a particular lifecycle/implementation method (Galvan, Mora, O’Connor, Acosta & Alvarez, 2015). The ISO/IEC 29110 basic profile is suitable for completing tasks regarding activities using different philosophies, depending on the VSEs needs. Due to the lack of established procedures and techniques, VSEs (especially in first-time implementations) have insufficient guidance about how such a profile should be implemented to fully achieve it (Sanchez, O’Connor & Colomo, 2015; Wongsai, Siddoo & Wetprasit, 2015), triggering incomplete execution of the standard, or even total withdrawal of the implementation. Delivery delays, cost overrun, over-budgeted human resources, and other negative factors in the development of software projects for VSEs are reasons for the wrong elicitation of requirements in early stages. By handling the basic profile processes and activities in an unstructured way, VSEs fail to perceive competitive advantages such as time reduction, standardized documentation, improvement in service and productivity, budget optimization, etc.

In this M.Sc. Thesis we propose a method for eliciting requirements in the context of the ISO/IEC 29110 basic profile based on executable pre-conceptual schemas. The proposed approach includes work products related to requirements elicitation in the project management and software implementation processes specified in the basic profile. By using executable pre-conceptual schemas for modeling such work products and executing their tasks, we develop an interactive map in order to guide VSEs when implementing the basic profile in early stages. Besides, by conducting a case study, we validate the proposed method. The case study is based on a research project financed by a governmental entity.

With this case study, we aim to evaluate the viability, effectiveness, and customer response of the method when performing the requirements elicitation of the project.
The proposed solution prioritizes the requirements elicitation in software development projects, enhancing accuracy, avoiding ambiguity, and easing communication between the work team and the customer. Besides, we intend to provide a resource to VSEs when implementing the basic profile of the ISO/IEC 29110 with the implementation of the lightweight method proposed in this M.Sc. Thesis.

This M.Sc. Thesis is organized as follows. In Chapter 1, we introduce the theoretical framework, which contains a summary of the ISO/IEC 29110 standard and pre-conceptual schemas. In Chapter 2, we review the state of the art. In Chapter 3, we define the problem statement by including the research question and the objectives. In Chapter 4, we propose a pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110. In Chapter 5, we validate the proposed model by conducting a case study. Finally, in Chapter 6 we discuss conclusions and future work.

1 Theoretical framework

1.1 ISO/IEC 29110 standard

The ISO/IEC 29110 standard “Lifecycle profiles for Very Small Entities” is a system project development guide aimed to improve VSE process in terms of economic benefits, time, efficiency, and documentation. Such a guide is intended to meet the VSE emerging need for competence, adaptability, and even survival in presence of an increasingly dependency on software. A VSE is an enterprise, organization, department or project of up to 25 people, considered “an entity that engages in systems or software engineering activities at any point, including development, integration, or maintenance” (ISO/IEC, 2016). Laporte, Séguin, Boas, and Buasung (2013) expose three main facts related to the creation of a standard for VSEs:

- Give VSEs a way to be recognized for producing quality software systems,
- Produce a set of standards and provide guidance to VSEs in establishing software engineering processes,
- Produce guides easy to understand, short, simple, and readily usable by VSEs.

According to Laporte, Alexandre, and O’Connor (2008), managers consider very important to be evaluated/certified with a standard in 74% of VSEs.

ISO/IEC 29110 standard includes five parts as illustrated in Table 1-1. Each part targets a specific audience. However, part 5 (management and engineering guide) incorporates a generic profile group for developing noncritical software, which applies to the majority of VSEs.

Table 1-1: ISO/IEC 29110 target audience (ISO/IEC, 2016).

ISO/IEC 29110	**Title**	**Target audience**
Part 1	Overview	VSEs, assessors, standards producers, tool vendors, and methodology vendors
Part 2	Framework and taxonomy	Standards producers, tool vendors, and methodology vendors. Not intended for VSEs
Part 3	Assessment guide	Assessors and VSEs
Part 4	Profile specifications	Standard producers, tool vendors, and methodology vendors. Not intended for VSEs
Part 5	Management and engineering guide	VSEs

The generic profile group comprises a four-profile collection: entry, basic, intermediate, and advanced. Each profile has additional progressive activities and elements for serving most VSEs.

- Entry profile. For start-up VSEs and small projects.
- Basic profile. For the development of a single project by a single team at a time.
- Intermediate profile. For VSEs involved in the development of more than one project in parallel with more than one work team.
- Advanced profile. For VSEs which want to sustain and grow as an independent competitive system development business.

1.1.1 Basic profile

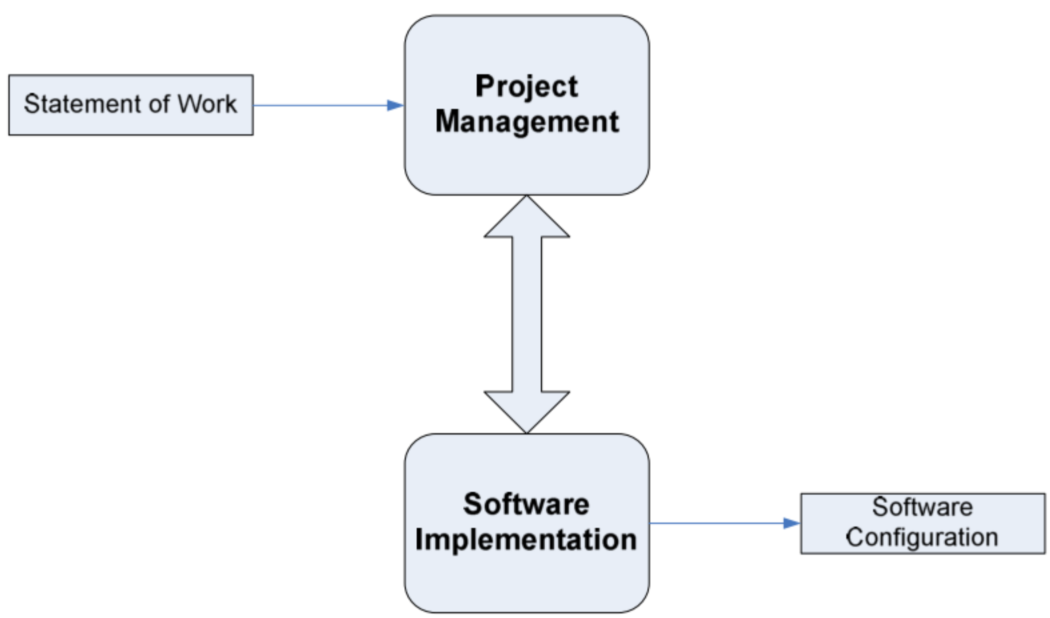
The basic profile integrates practices based on the selection of ISO/IEC 12207. According to the ISO/IEC (2011), the main benefits obtained by the implementation of such profile includes:

- Development of a collection of project requirements and expected products.

- Presence of discipline in activities, correctness capability, and major visibility of the project due to process management.
- Satisfaction of the acquirer needs and quality product assurance by doing systematic software implementation process.

The basic profile is applicable to VSE projects up to 25 people. Such a profile is suitable for the development of a single noncritical system at a time. The project could fulfill an external or internal contract. Likewise, the basic profile can operate in conjunction with any development approach such as evolutionary, agile, incremental, waterfall, iterative, etc. The basic profile includes two main processes: project management and software implementation. The ISO/IEC 29110 is independent to any method for executing activities described in the two processes. Besides, such a profile includes a set of documents for accomplishing process implementation and a set of roles responsible for task execution. Both processes are interrelated as seen in Figure 1-1. According to the ISO/IEC (2011), the project management process helps to identify, determine, and carry out the tasks to be performed, complying with the project objectives in the expected quality, time, and cost. On the other hand, the software implementation process helps to develop the analysis, design, construction, integration, and test activities for software products according to the specified requirements.

Figure 1-1: Basic profile guide processes (ISO/IEC, 2011).



1.1.2 Project management

Project management process includes seven objectives, which are related to the ISO/IEC 12207:

- Develop the project plan according to the statement of work reviewed and accepted by the customer.
- Monitor the project progress against the project plan and record the actual condition in the progress status record.
- Review and analyze the change requests.
- Coordinate meetings with the work team and the customer.
- Identify risks along the project.
- Establish a software version strategy.
- Guarantee software quality.

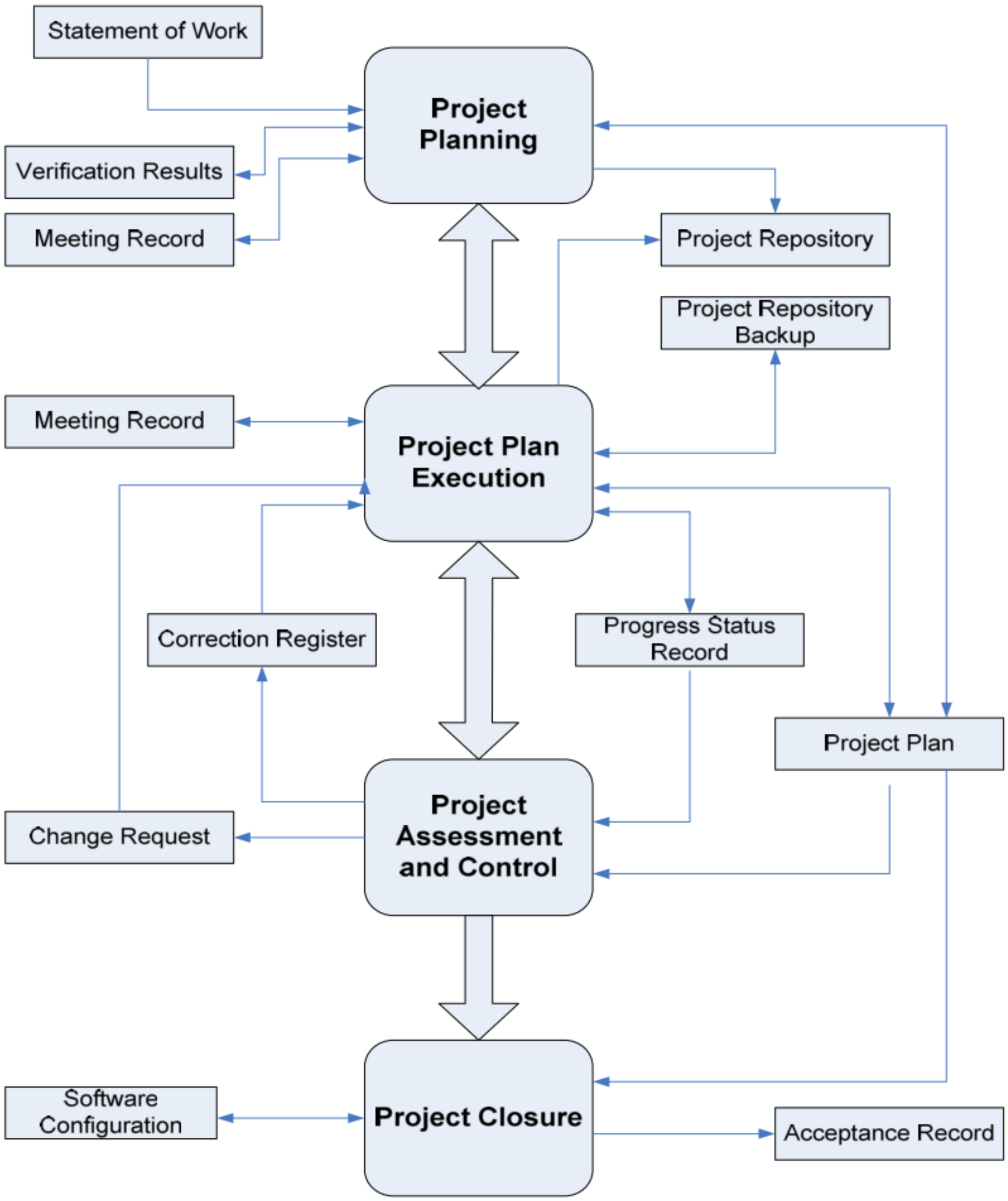
Roles involved in task execution for the project management process are presented in Table 1-2.

Table 1-2: Project management roles involved (ISO/IEC, 2011).

Role	**Abbreviation**
Customer	CUS
Project Manager	PM
Technical Leader	TL
Work Team	WT

The ISO/IEC 29110 includes a set of four activities for the project management process: project planning, project plan execution, project assessment and control, and project closure. Each activity includes tasks to be performed by the defined roles. Moreover, the development of each task demands an input work product as mandatory, and the completion of such task returns an output work product. In Figure 1-2, we illustrate the project management process diagram, including the main activities and related work products.

Figure 1-2: Project management process diagram (ISO/IEC, 2011).



The general requirements are mostly determined, performed, and registered in the statement of work, which is the first input work product of the project management process. Nevertheless, some tasks necessary for the requirements elicitation are developed in the project plan. The statement of work is an input work product for the project planning activity as observed in Figure 1-2,

1.1.2.1 Statement of work

Statement of work is an entry condition, which needs to be fulfilled and documented by the customer in order to start the project management process. The statement of work is the first work product and is completed and reviewed by the technical leader and the project manager for elaborating the project plan. According to the ISO/IEC (2011), the statement of work includes the product description, purpose, general customer requirements, scope, objectives, contract deliverables with their software components, and the customer requirements (See Table 1-3).

Table 1-3: Statement of work description (ISO/IEC, 2011).

Name	Description	Source

Statement of Work	Description of work to be done related to *Software* development. It may Include:
	
	- *Product Description*
	  - Purpose
	  - General Customer requirements
	- *Scope* description of what is included and what is not
	- *Objectives* of the project
	- *Deliverables* list of products to be delivered to Customer
	
	The applicable status is: reviewed.
		Customer

1.1.2.2 Project plan

Activities and tasks to be performed in order to assure the project completion are specified in the project plan. The project plan includes a set of work products coordinated, reviewed and approved by the project manager, accepted by the customer, and executed by the team. The statement of work is the input work product on which the project plan is based. The development of the project plan starts in the project planning process. In the project plan execution process, such a plan is monitored with the progress of the project and the results are recorded in the progress status record. In the project assessment and control process, the project manager evaluates, executes corrective actions, and performs changes in the project plan according to the progress status record. Finally, in the project closure process, the delivery instructions established in the project plan are formalized and the acceptance record is signed. In Table 1-4 we show a description of each component included in the project plan.

Table 1-4: Project plan description (ISO/IEC, 2011).

Name	Description	Source

Project Plan	Presents how the project processes and activities will be executed to assure the project's successful completion, and the quality of the deliverable products. It Includes the following elements which may have the characteristics as follows:
	
	- *Product Description*
	    - Purpose
	    - General Customer requirements
	- *Scope* description of what is included and what is not
	- *Objectives* of the project
	- *Deliverables* - list of products to be delivered to Customer
	- *Tasks*, including verification, validation and reviews with Customer and Work Team, to assure the quality of work products. *Tasks* may be represented as a Work Breakdown Structure (WBS).
	- *Estimated Duration* of tasks
	- *Resources* (humans, materials, standards, equipment and tools) including the required training, and the schedule when the *Resources* are needed.
	- *Composition of Work Team*
	- *Schedule of the Project Tasks,* the expected start and completion date for each task, and the relationship and dependencies of the *Tasks*.
	- *Estimated Effort and Cost*
	- *Identification of Project Risks*
	- *Version Control Strategy*
	    - Product repository tools or mechanism identified
	    - Location and access mechanisms for the repository specified
	    - Version identification and control defined
	    - Backup and recovery mechanisms defined
	    - Storage, handling and delivery (including archival and retrieval) mechanisms specified
	- *Delivery Instructions*
	    - Elements required for product release identified (i.e., hardware, software, documentation etc.)
	    - Delivery requirements
	    - Sequential ordering of *Tasks* to be performed
	    - Applicable releases identified
	    - Identifies all delivered *Software Components* with version information
	    - Identifies any necessary backup and recovery procedures
	
	The applicable statuses are: verified, accepted, updated and reviewed.
		Project Management

1.1.3 Software Implementation

According to the ISO/IEC (2011), the software implementation process includes seven objectives:

- Accomplish task of the activities detailed in the project plan.
- Define, analyze for correctness and testability, and obtain customer approval of the software requirements elicited in the project management process.
- Develop software architecture and design, describe software components and the interfaces related, and establish consistency and traceability to software requirements.
- Produce software components and define unit tests.
- Perform integration of software components, verify such components by using test cases and test procedures, correct the defects, and record the test report.
- Elaborate the software configuration including user, operation, and maintenance documentation.
- Verify and validate tasks related to all work products.

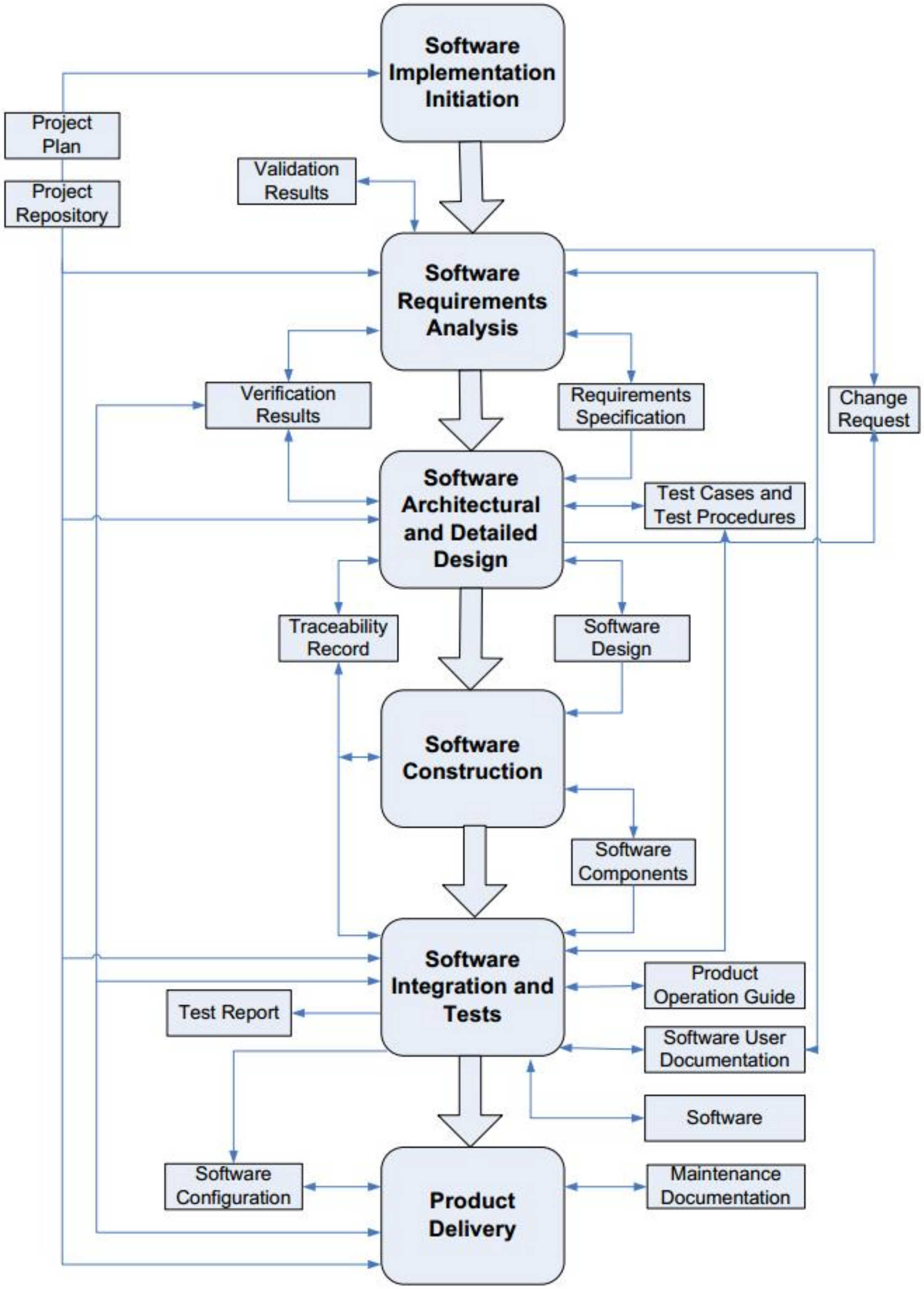
Roles involved in the software implementation process are presented in Table 1-5.

Table 1-5: Roles involved in the software implementation process (ISO/IEC, 2011).

Role	**Abbreviation**
Customer	CUS
Analyst	AN
Designer	DES
Programmer	PR
Project Manager	PM
Technical Leader	TL
Work Team	WT

As illustrated in Figure 1-3, the software implementation process includes six activities: software implementation initiation, software requirements analysis, software architectural and detailed design, software construction, software integration and tests, and product delivery (ISO/IEC, 2011).

Figure 1-3: Software implementation process diagram (ISO/IEC, 2011).



1.1.3.1 Requirements specification

The software requirements analysis activity has as output work product the complete requirements specification. Such an activity includes tasks regarding analysis and specification of customer requirements, agreement on the customer requirements, verification and validation of requirements, assignment of task described in the project plan, etc. Such a work product includes a set of sub-work products as detailed in Table 1-6.

Table 1-6: Requirements specification structure (ISO/IEC, 2011).

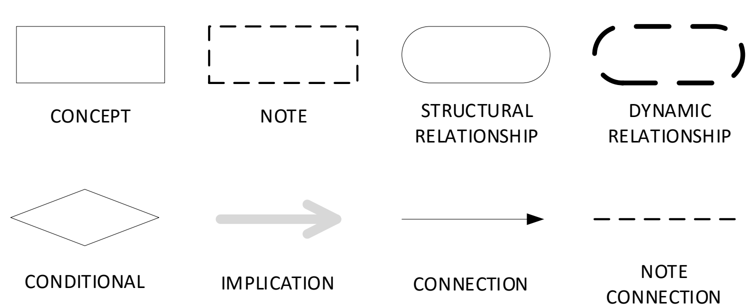
Name	Description	Source

Requirements Specification	Identifies the software requirements. It may have the following characteristics:
	
	- Introduction –general description of *Software* and its use within the *Scope* of the Customer business;
	- Requirements description:
	  - Functionality – established needs to be satisfied by the *Software* when it is used in specific conditions. Functionality must be adequate, accurate and safe
	  - User interface – definition of those user interface characteristics that allow to understand and learn the *Software* easily so the user be able to perform his/her *Tasks* efficiently including the interface exemplar description
	  - External interfaces – definition of interfaces with other software or hardware
	  - Reliability – specification of the software execution level concerning the maturity, fault tolerance and recovery
	  - Efficiency – specification of the software execution level concerning the time and use of the *Resources*
	  - Maintenance – description of the elements facilitating the understanding and execution of the future *Software* modifications
	  - Portability – description of the *Software* characteristics that allow its transfer from one place to other
	  - Design and construction limitations/constraints – needs imposed by the Customer
	  - Interoperability – capability for two or more systems or *Software Components* be able to change information each other and use it
	  - Reusability – feature of any product/sub-product, or a part of it, so that it can be used by several users as an end product, in the own software development, or in the execution of other software products
	  - Legal and regulative – needs imposed by laws, regulations, etc.
	
	Each requirement is identified, unique and it is verifiable or can be assessed.
	
	The applicable statuses are: verified, validated and baselined.
		Software Implementation

1.2 Pre-conceptual schemas

A pre-conceptual schema is a knowledge representation created in 2006. According to Zapata, Gelbukh, and Arango (2006a), a pre-conceptual schema is "a simple representation framework for ontological knowledge with dynamic and deontic characteristics." Such schemas are aimed to transform used-common language into modeling-structure language (Zapata, 2007a). By using pre-conceptual schemas, we can represent natural language discourse of a certain problem, process, or domain in a graphical environment (Manrique & Zapata, 2013). Besides, the construction of pre-conceptual schemas helps to reduce the speech gap between stakeholders and analyst, increasing the probability of an achievement approach when translating from natural to modeling language. Pre-conceptual schemas are used to express a domain terminology representation by using models, in order to ease translation to conceptual schemas. Likewise, pre-conceptual schemas help to map verbal expression of a problem defined by a stakeholder into an unambiguous graph, which can be modeled and implemented by a development team (Zapata, Lezcano & Tamayo, 2007b). According to Zapata and Arango (2005), the symbolic graphical nature of a pre-conceptual schema grants a clear interpretation to analysts and developers with a minimum training. By using pre-conceptual schemas, we can obtain several UML (Unified Modeling Language) diagrams such as class, communication, and state machine (Zapata, Gelbukh & Arango, 2006a). Besides, pre-conceptual schemas can be executable, allowing stakeholder for understanding system instances and operational behavior by showing the functionality of the modeled domain, and acting as an interactive and agile road map (Zapata, Giraldo & Londoño, 2011). In addition, such schemas have several useful features, including integration of concepts, dynamic elements, and proximity to the stakeholder language (Zapata, Gelbukh & Arango, 2006a). The basic syntax of pre-conceptual schemas is illustrated in Figure 1-4.

Figure 1-4: Basic syntax of pre-conceptual schemas (Zapata & Vargas, 2013).

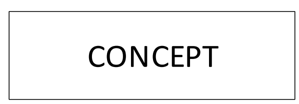


1.2.1 Syntactic elements of a pre-conceptual schema

1.2.1.1 Concept

Concepts represent nouns or noun phrases cited in the stakeholder discourse (Zapata, Gelbukh & Arango, 2006a). Such nouns could be actors—*e.g.,* secretary, teacher, assistant—, unanimated things—*e.g.*, invoice, office, chair, tree—and attributes—*e.g.*, age, name, type, color, size. Likewise, noun phrases such as “high school teacher,” “cellular phone,” etc. can be used as concepts. In Figure 1-5, we illustrate the representation of a concept.

Figure 1-5: Graphical representation of a concept (Zapata & Vargas, 2013).



1.2.1.2 Structural relationship

Structural relationships are represented by using a continuous-line oval (See Figure 1-6), with the purpose of linking two concepts permanently. Structural relationships could be labeled with two verbs: be or have. The union of two concepts by means of a structural relationship is called structural triad as shown in Figure 1-7 (Zapata, 2007a).

Figure 1-6: Graphical representation of a structural relationship (Zapata & Vargas, 2013).

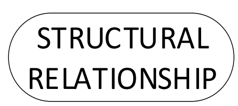
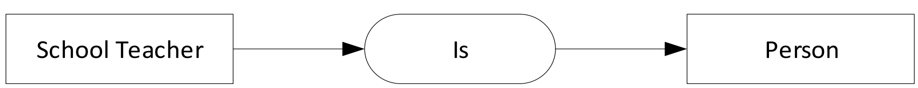


Figure 1-7: Example of a structural triad (The Authors).



1.2.1.3 Dynamic relationship

According to Zapata (2007a), a dynamic relationship express an action verb—*e.g.*, fill, save, copy, develop—represented by a dotted line-oval (See Figure 1-8). In Figure 1-9, we show a dynamic triad, which is the result of using a dynamic relationship by linking two concepts.

Figure 1-8: Graphical representation of a dynamic relationship (Zapata & Vargas, 2013).

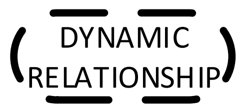
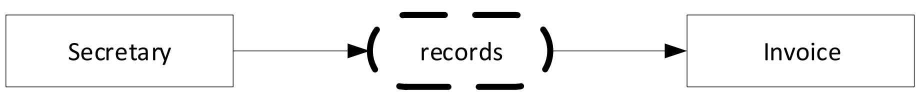


Figure 1-9: Example of a dynamic triad (The Authors).



1.2.1.4 Connection

Connections are elements represented by an arrow (See Figure 1-10), with the purpose of linking concepts with relationships and *vice versa* (Zapata, 2007a).

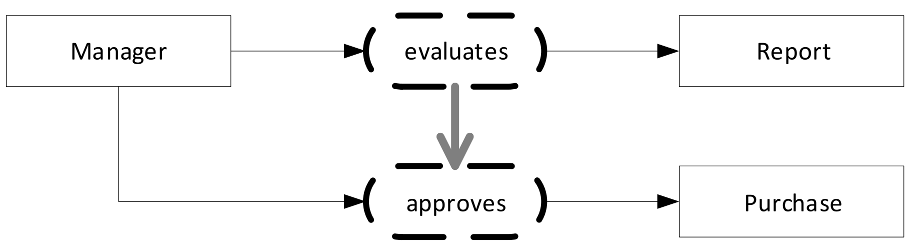
Figure 1-10: Graphical representation of a connection (Zapata & Vargas, 2013).

1.2.1.5 Implication

An implication is a cause-effect relationship between dynamic relationships, which makes the first dynamic relationship to act as an antecedent and the second dynamic relationship as a consequent (Zapata, 2007a). “These elements can be only connected from one relationship to another, and it means the target verb it is only performed if the source verb is performed” (Zapata, Gelbukh & Arango, 2006b). An implication is represented by using thick grey arrows as shown in Figure 1-11. Finally, in Figure 1-12 we show a cause-effect example for using such an element between two dynamic relationships.

Figure 1-11: Graphical representation of an implication (Zapata & Vargas, 2013).

Figure 1-12: Example of an implication (The Authors).



1.2.1.6 Conditional

Conditional elements represent the different causality relationships in a domain. Such elements specify the concept restrictions and act as logical preconditions for fulfilling a dynamic relationship (Zapata, Gelbukh & Arango, 2006a). Restrictions expressed in conditionals are composed by a combination of concepts and algebraic/logic operators. As we show in Figure 1-13, a rhomb represents a conditional element. Besides, in Figure 1-14, we illustrate an example of using a conditional in a dynamic triad.

Figure 1-13: Graphical representation of a conditional (Zapata & Vargas, 2013).

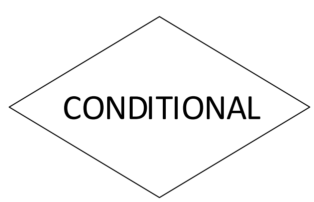
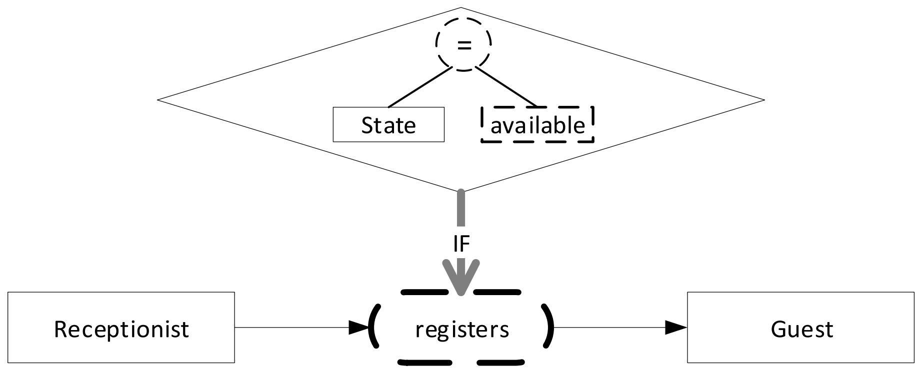


Figure 1-14: Example of a conditional (The Authors).



1.2.1.7 Reference

A reference is a numbered circle, which links physically distant concepts/relations in the diagram (See Figure 1-15; Zapata, 2007a). In Figure 1-16 we show an example of such an element.

Figure 1-15: Graphical representation of a reference (Hidalgo, 2009).

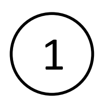
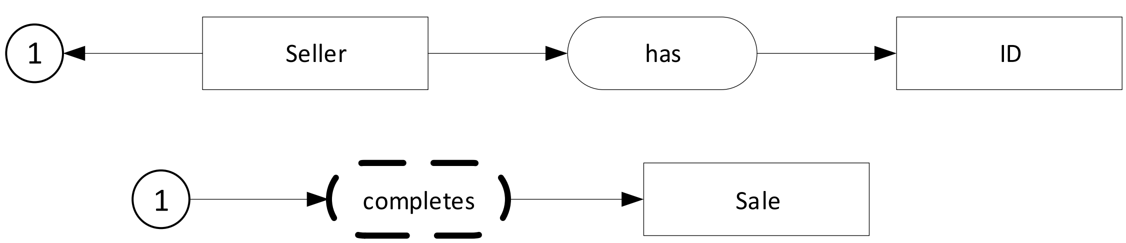


Figure 1-16: Example of a reference (The Authors).



1.2.1.8 Note

Notes are dotted rectangles with the purpose of defining possible values of concepts (See Figure 1-17; Zapata, Gelbukh & Arango, 2006a). A dotted line is a connection between concepts and notes as seen in the example illustrated in Figure 1-18.

Figure 1-17: Graphical representation of a note (Zapata & Vargas, 2013).

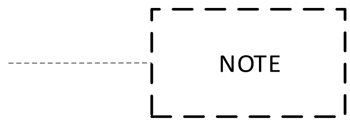
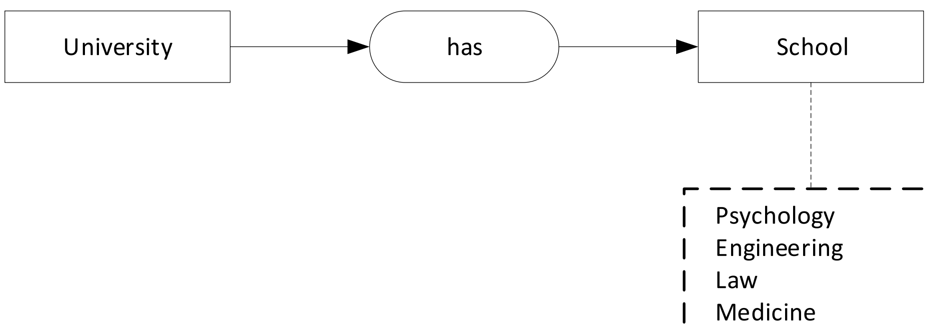


Figure 1-18: Example of a note (The Authors).



2 State of the art

2.1 Process improvement in VSEs

Process improvement is an ongoing practice comprising a set of proactive tasks for evaluating, diagnosing, and enhancing business processes in order to meet new quality standards, reduce development time, reduce cost, improve productivity, implement a standard to meet market regulation, implement new tools or new methodologies, etc. Process improvement involves a specific method in order to increase effectiveness and/or efficiency for meeting organizational goals. Main results of process improvement can be measured in customer satisfaction, employee and supplier relationship, business results, product and service quality, productivity and profit, expenses, among others (Laporte & O’Connor, 2016). A weakness VSEs face in the globalized world is the lack of standardization of internal processes, impeding alignment to the company objectives and goals. Lack of structured methods in VSEs could negatively influence the resulting product or service, making difficult for companies to obtain maximum benefit from their resources—*e.g.* real estate, furniture, economic, human resources, etc. The Organization for Economic Co-operation and Development (OECD) evaluated the amount of SME (small and medium-sized enterprises). According to OECD (2005), "SMEs constitute the dominant form of business organisation in all countries world-wide, accounting for over 95% and up to 99% of the business population depending on country." Software standards like CMMI (Capability Maturity Model Integration) and ISO 12207 are applicable to enterprises with all sizes. However, due to their extensive implementation and cost, implementing such standards in VSEs is very difficult (Siddoo, Wongsai & Wetprasit, 2013). ISO/IEC 29110 standard “Lifecycle profiles for Very Small Entities” was created for guiding VSEs to software processes standardization, with profiles defined for project management and software implementation in order to develop sequential and parallel activities (ISO/IEC, 2011).

2.2 Case studies

Previous case studies have been developed in companies located in several countries around the world. The identified case studies are presented as follows:

2.2.1 The Development and Experimentation of an International Standard for Very Small Entities Involved in Software Development

Laporte (2011) elaborates a study characterizing initial pilot projects for implementing the ISO/IEC 29110. Such projects are intended to encourage the standard adoption in VSEs. Besides, the exercise is aimed to provide tailorable and viable guidelines for successfully conducting projects in VSEs. In Table 2-1 we show the pilot project classified by country and type.

Table 2-1: ISO/IEC 29110 pilot projects (Laporte, 2011).

VSE	Type	Country
Computer Aided Design (CAD) Software Support Organisation	Completed	Canada
School Board of the Montréal Area	Completed	Canada
Software Engineering Graduate students SQACourse: • Insurance Company • Support Organisation for Notaries • Geographic Information System Modeling Company • Support Organisation for Lawyers • University Research Laboratory • Acme Software for Building Maintenance • Acme Insurance • Acme Security • Acme Web Site Development • Acme Communications	Completed	Canada
CETIC	Underway	Belgium
UBO	Underway	France
LERO	Underway	Ireland
Research Laboratory in Medical Imagery and Orthopedic	Underway	Canada
Telecommunication Research Chair	Underway	Canada

Although case studies included in this paper belong to different countries and organizations, they lack a clear specification on how to implement the standard. Likewise, the aim of pilot projects included in this study is reduced to evaluate the viability, performance, and response of VSEs when introducing the ISO/IEC 29110 in their work philosophy. Therefore, the findings of this study serve as the basis for elaborating methods in order to apply the standard.

2.2.2 Implementing Process Improvement in Very Small Enterprises with ISO/IEC 29110: A Multiple Case Study Analysis

Laporte and O’Connor (2016) present the details of nine case studies involving pilot usages of the ISO/IEC 29110 basic profile. The authors communicate the success stories from pilot trials of the standard adoption. Besides, the lessons learned in the case studies should assist VSEs when implementing the basic profile. The case studies presented in this study are listed as follows:

- Case 1: An IT Start-up
- Case 2: A Canadian/Tunisian IT start-up
- Case 3: A large IT consulting firm
- Case 4: A large Canadian financial institution
- Case 5: A division of a large Engineering Enterprise
- Case 6: An Automotive Enterprise
- Case 7: A large electricity provider
- Case 8: A medical R&D VSE
- Case 9: A young transportation company

In case study 9, the authors used the systems engineering basic profile. According to Laporte and O’Connor (2016), “ISO 29110 was a good starting point to align processes with selected level 2 and 3 practices of the CMMI model.” In Table 2-2, we illustrate the percentage of coverage of the company processes to CMMI-DEV.

Table 2-2: Coverage of the company processes to CMMI-DEV (Laporte & O'Connor, 2016).

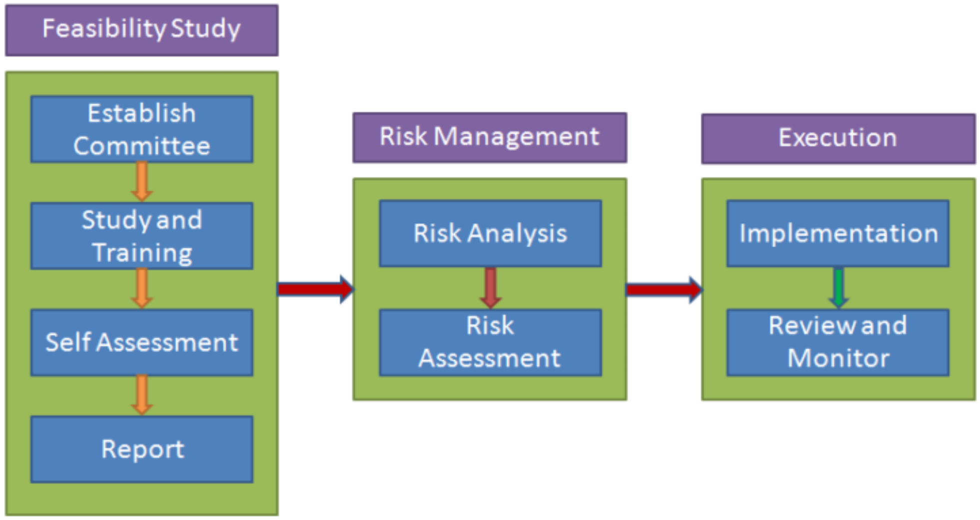
CMMI Level 2 Process Areas	Percentage of Coverage
Configuration Management	50-70%
Measure and Analysis	20-40%
Project Monitoring and Control	70-90%
Project Planning	70-90%
Process and Product Quality Assurance	45-65%
Requirements Management	90-100%
Supplier Agreement Management	70-90%

After completing the case studies, Laporte and O'Connor (2016) demonstrate the viability of the activities and tasks defined in the ISO/IEC 29110 for improving quality in software projects. Nevertheless, as the authors expose, VSEs include competent technical people, but lack people trained in project management. Therefore, the inclusion of an easy, lightweight, and viable method is needed. Due to the lack of established procedures and techniques, VSEs have insufficient guidance about how such a standard should be implemented and fully achieved.

2.2.3 An Implementation Approach of ISO/IEC 29110 for Government Organizations

Siddoo, Wongsai, and Wetprasit (2013) review an implementation approach of the ISO/IEC 29110 in a small information technology department in a Thai government academic institute. Faculty of Technology and Environment, Prince of Songkla University, Phuket Campus is a government agency with an in-house development entity composed by a group of 4-5 developers. According to Siddoo, Wongsai, and Wetprasit (2013), "the group of developers never completely applied software engineering and project management disciplines into the software development projects". Consequently, the unit adopts the basic profile of the ISO/IEC 29110 for process enhancing. After consulting software process improvement experts, the development team came up with a 3-step approach as illustrated in Figure 2-1.

Figure 2-1: Implementation approach (Siddoo, Wongsai & Wetprasit, 2013).



The authors established the statement of work as the starting point of the ISO implementation, including initial user requirements and work commitment in such a work product. However, the proposed approach lacks guidelines for requirements elicitation in early stages of development. In addition, risk management (risk analysis and risk assessment) is a project management activity, directly dependent on a correct requirements elicitation in the statement of work. Therefore, without a solid statement of work defined, all activities exposed in the solution are subject to failure.

2.2.4 Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru

García, Laporte, Arteaga, and Bruggmann (2015) introduce the implementation of the ISO/IEC 29110 standard in a four-person IT startup company in Peru. The implementation was carried out in an agile environment, took about 900 hours, and spend 18% of the total project effort for implementing the standard. Besides, the authors describe how the VSE became the first Peruvian company to which an ISO/IEC 29110 certificate of conformity was granted. The management and engineering guide of the ISO/IEC 29110 basic profile is unrelated to a particular implementation method (ISO/IEC, 2011). Therefore, the VSE had its own templates for each work product. Likewise, the VSE performed evaluation criteria and weights of the PM and SI processes of ISO/IEC 29110 in order to analyze risks. Finally, in Table 2-3 we show an analysis performed to verify VSE met the entry conditions in order to accomplish the basic profile.

Table 2-3: Examples of entry conditions and results obtained in the evaluation of the VSE (García, Laporte, Arteaga & Bruggmann, 2015).

Assessment question entry condition	Response obtained
Is there a project contract with statement of work?	The VSE has two ways to establish a contract: 1. The client contacts the VSE. If it is a valid opportunity, a draft version of the contract is created. The contract is updated with the agreements reached as a result of negotiations between the VSE and the client. This contract becomes the statement of work (SoW). 2. The VSE is submitted to a tender of the Organismo Supervisor de las Contrataciones del Estado Peruano (Supervisory Board of Peruvian Government Procurement [OSCE]). The SoW is the Terms of Reference (ToR) document, in which the scope, duration, and requirements of the project are determined.
Are the feasibility of cost, schedule, and technical aspects evaluated before the project starts?	When the customer contacts the VSE, the terms for the schedule, budget, and technologies used in the project are agreed during negotiations. In case of open call for tenders by the OSCE, the ToR document contains the costs and technical requirements needed for the project. It also established the set duration of the project and major milestones of project deliverables.

The basic profile included in the ISO/IEC 29110 is intended to guide VSEs in order to implement the "best practices" when developing system projects. Therefore, such a profile exclude methods or tools for developing activities and tasks listed in the standard processes. Consequently, even though a company is certified in ISO/IEC 29110, a selection of unproductive methods is still possible, leading to unexpected results. The VSE included by García, Laporte, Arteaga, and Bruggmann (2015) has activities as a list of responses to an assessment question entry condition, lacking the use of knowledge representations. The project plan is the core of the ISO/IEC 29110 and entirely depends on the requirements elicitation performed in the statement of work. Therefore, the use of knowledge representations at this stage of implementation is necessary in order to avoid ambiguity, delays, cost overruns, and misunderstandings.

2.2.5 ISO/IEC 29110 Implementation on two Very Small Software Development Companies in Lima

Díaz, de Jesús, Melendez, and Dávila (2016) identify a set of lessons learned as a result of the implementation of the ISO/IEC 29110 basic profile in two VSEs in Lima. Involved companies in the study were named *Lim.Epsilon* and *Lim.Nu.* In order to preserve the confidentiality. The authors performed a diagnosis evaluation for determining the VSEs capability for process improvement (See Table 2-4).

Table 2-4: Business problems identified in companies (Translated from Díaz, de Jesús, Melendez & Dávila, 2016).

Id	Business problems		
	Description	**Lim. Epsilon**	**Lim. Nu.**
1	Inefficient or non-existent communication among work team members about assigned tasks and their relations	X	X
2	Erroneous work time estimation of the assigned tasks	X	X
3	Low levels of products and services marketing	X	X
4	Weak incorporation of international practices	X	X
5	Lack of documentation for executing processes	X	X
6	Few project control activities comparing planned processes vs executed processes	---	X
7	High learning curve in implementing used technologies and emerging technologies	X	---

Lessons learned from this study reveal the importance of meetings between company staff and work team. In addition, support of the organization Chief Executive Officer (CEO) is fundamental for process improvement implementation. Initial evaluation regarding business problems helps to plan and execute effective improvement proposals. Constant task monitoring allows for verifying the project progress, contributing to the correct development of improvement proposals. Besides, software engineering techniques and support tools are necessary to ease process improvement (Díaz, de Jesús, Melendez & Dávila, 2016). Consequently, graph-like representations should be included as software engineering methods for process improvement, enhancing success probabilities for implementing standards. The manifestation of the ISO/IEC 29110 basic profile domain by using knowledge representations, allows VSEs to perform the standard adoption in an intuitive and unambiguous way, incrementing viability when implementing such a profile.

2.2.6 Implementation of the Basic Profile of ISO/IEC 29110 in a Small Software Developer Company: Lessons Learned

Abarca, Arisaca, and Dávila (2015) present the experience for implementing the basic profile of the ISO/IEC 29110 in a small software development company located in Arequipa-Peru. Main results of this study include a list of identified problems and actions taken (see Table 2-5), as well as lessons learned from the process. From the effort and process improvement in the VSE, Abarca, Arisaca, and Dávila (2015) identify a set of lessons learned. The authors highlight the contribution performed by internal and external auditors to detect deficiencies in the process. Likewise, by implementing internal training, the staff was able to understand and unify the model criteria. Finally, by standardizing documentation, the company was capable of managing the process centralized. However, the lack of knowledge-representation-based methods used for implementing the standard cause the process to have relatively high complexity level, which in long run generates more implementation time than necessary.

Table 2-5: Problems and solutions of the VSE (Translated from Abarca, Arisaca & Dávila, 2015).

Problem	Type	Solution
The staff shows rejection for corresponding tasks of the new framework.	Participation	Create an organizational culture, through continuous supervision.
The process improvement requires additional human resources to align the organization goals and needs.	Backup	Assignee an employee to monitor goals fulfillment.
Processes are very time consuming for team members.	Work practices	Equitable distribution of work.
Lack of communication of needs or inconveniences.	Communication	Weekly communication meetings for each project.
The changes and tasks assigned do not show short-term improvements.	Work practices	Train the staff about the tasks to be performed as well as the benefits.
Unreal expectations regarding the improvement.	Change management	Personal training.
Lack of metrics reflecting the improvement effect.	Assessment	Comparison delivery time and customer satisfaction with similar projects.
Lack of risk management.	Change management	Review of events in past projects identifying possible risks for the new framework.

In a case study described by O´Connor (2014) of an early adopter of ISO/IEC 29110 project management practices, a series of post-mortem interviews are conducted at the project end. In an extract from an interview a CEO says “they [ISO] want us [VSEs in general] to adopt standards then they should make it easier for us… they should give us complete how to guides and not just a list of task criteria… its too long and too difficult to create all these processes” (O´Connor, 2014). By following the perspective declared by the company CEO, the standard implementation should be easy and fast. Simple and smooth implementation methods are needed to accomplish a successful and agile process. Knowledge representations ease translation from natural language and domain environments into concept representations. Such a fact influences the amount of time and effort spent on a specific topic and generates a positive gap. In addition, by using knowledge representations from early stages in the implementation of the ISO/IEC 29110 basic profile, both learning curve and time invested in the implementation of the standard could be developed in short periods.

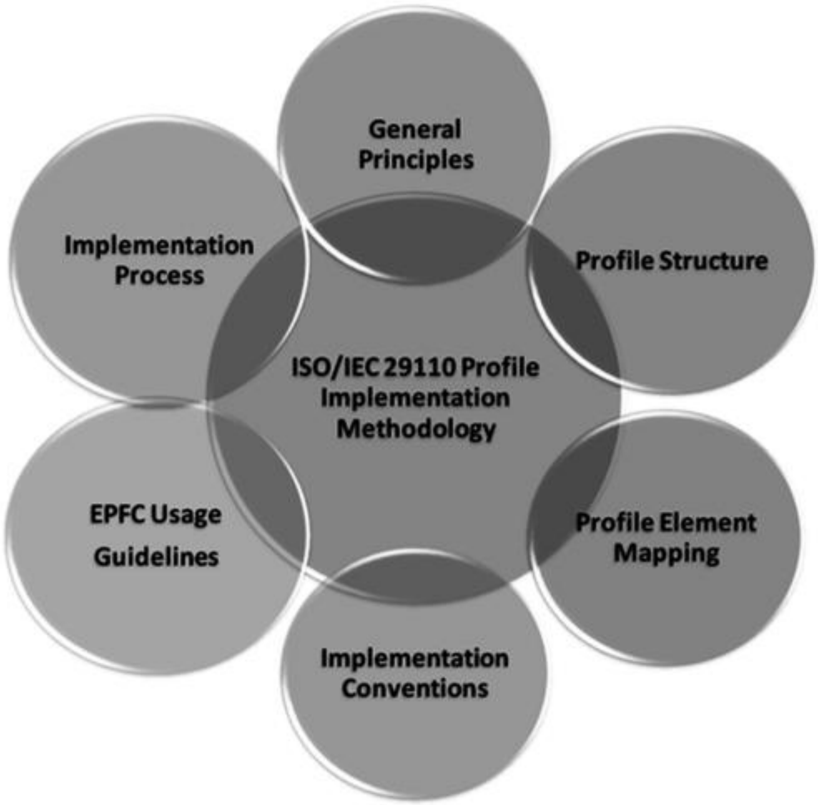
2.3 Models for implementing the ISO/IEC 29110 basic profile

Previous approaches have been developed in order to support a viable ISO/IEC 29110 implementation.

2.3.1 Methodology for ISO/IEC 29110 Profile Implementation in EPF Composer

Buchalcevova (2017) presents a method developed to manage the implementation of the ISO/IEC 29110 entry and basic profiles in the open-source content management tool Eclipse Process Framework (EPF) composer. Such a development is published in the form of a web application and it is used for educational and research purposes in the Czech Republic. As illustrated in Figure 2-2, the method proposed includes six elements: general principles, profile structure, profile element mapping, implementation conventions, EPF composer usage guidelines, and implementation process.

Figure 2-2: ISO/IEC 29110 Profile implementation methodology structure (Buchalcevova, 2017).



General principles include a set of ten principles for implementing the VSE profiles in the EPF composer (Buchalcevova, 2017):

- Main objects are created in the framework—*e.g.*, pages, images, tables, etc.
- The software content categories are mapped into the profile processes.
- Multiple views for reviewing the profile from various perspectives—*e.g.*, goals, roles, work products, etc.—are developed.
- Hyperlinks are used.
- Categories created should be used.
- Digital profile version should include all information in printed version.
- Digital and printed version of the profile should correspond each other.
- Digital and printed version of the profile should have the same format.
- Digital and printed version of the profile should have the same order of elements.
- Duplicity for capturing relations between the elements is used.

In the profile structure process, the author defines a set of categories and pages covering the entire content. For example, the entry profile contains the following elements:

- Introduction
- Project Management Process
- Software Implementation Process
- Roles
- Work Products
- Software Tools

After categorizing the elements, a profile individual element mapping to EPF composer content categories is needed in order to accomplish the general principles. The author defines specific conversions while working with the EPF such as composer naming conversions, usage of predefined styles, hypertext content guidelines, and metadata management. Likewise, a set of practical recommendations and guidelines for the implementation of the profiles is developed. Finally, a text editing guideline is created in order to avoid problems such as text formatting. According to Buchalcevova (2017), “the profile in a printed form is a comprehensive text, which is quite difficult for the user to navigate through and lacks an emphasis on their clarity and coherence.” Therefore, the author proposes an approach for reviewing the ISO/IEC 29110 entry and basic profiles by using multiple views in order to evaluate different perspectives. However, the alternative views proposed exclude knowledge representations, preventing the user from being able to interact with the system in a friendly and graphical way. Knowledge representations allow users for fluently communicating with the system, leading to clarify the standard implementation process.

2.3.2 Ontology-based Tailoring of Software Process Models

Eito (2014) describes an ontology-based model to organize the collaboration framework of software development teams. The author presents the framework including three layers:

- The core layer, based on the ISO/IEC 29110 basic profile.
- The corporate layer, based on the individual company policies and work procedures.
- The project layer, “based on the result of tailoring the corporate layer to accommodate specific requirements imposed by the customer or the project’s stakeholders.” (Eito, 2014)

The resulting ontology was constructed by using the Web Ontology Language (OWL), which can be easily reused in other applications and serve to multiple purposes. Besides, by using such a language, the ontology integration with a semantic wiki tool is simpler. According to Eito (2014), “dealing with requirements coming from different sources implies a risk, as staff involved in project planning and execution may disregard relevant requirements and information…In addition, dealing with separate documents and standards has a negative impact on productivity, due to the difficulties to remember at a given time all the applicable constraints.” Therefore, the framework has a common access point to register and locate all requirements and documents involved in the process. Besides, Eito (2014) indicates the importance of requirements elicitation, expressing project staff “should know the impact that customer requirements and applicable standards have on their activities and on the product deliverables they need to produce, verify or validate.” Although ontologies are strong frameworks to represent domains, their relations, and concepts, they lack a visual process flow to navigate from a starting to a final point. Graphical knowledge representations allow for following activities and completing tasks in a visual manner (like following a map), leading to the finalization of a main process.

3 Problem statement

3.1 Research question

The ISO/IEC 29110 standard “Lifecycle profiles for Very Small Entities” was created to meet the VSEs need for developing quality software and improving time, budget, documentation, etc. The basic profile defined in the management and engineering guide of such a standard has been implemented by VSEs around the world. Such a lightweight process allows for the execution of one project at a time. The project management and software implementation processes established in the basic profile of the ISO/IEC 29110 include a set of activities and tasks accepted as best practices for improving quality in software projects. However, the basic profile is unrelated to a particular implementation method, aiming on the what-to-do but leaving the how-to-do-it choice at discretion of the organization willing to implement such a profile. Due to the lack of established procedures and techniques, VSEs (especially in first-time implementations) have insufficient guidance about how such a standard should be implemented to fully achieve it, triggering incomplete execution of the standard, or even total withdrawal of the implementation. Likewise, previous work is devoted to develop case studies such as pilot projects, ISO/IEC 29110 certifications, and partial/full implementations. Such case studies are aimed to evaluate the viability of the ISO/IEC 29110 when introducing the standard in VSEs, excluding the construction of a particular method when implementing such a standard. Also, they lack a comprehensive study of requirements elicitation when implementing the basic profile, even when an entire project is subject to failure when such an activity is inaccurate. Finally, previous methods constructed lack the inclusion of knowledge representations in order to perform the standard adoption in an intuitive and unambiguous way. Since the main core of the project management process is the project plan (which is entirely based on the requirements elicitation made with the acquirer and developed from the very start of the project until the project closure) a lightweight method based on modeling languages should be defined. An accurate requirements definition process should be the core of the method.

3.2 Objectives

3.2.1 General objective

Developing a method for eliciting requirements in very small organizations based on the ISO/IEC 29110 basic profile standard by using pre-conceptual schemas.

3.2.2 Specific objectives

- Characterizing requirements elicitation of the ISO/IEC 29110 basic profile.
- Defining the method for eliciting requirements.
- Constructing the pre-conceptual schema representing the requirements elicitation.
- Validating the method obtained by proposing a case study.

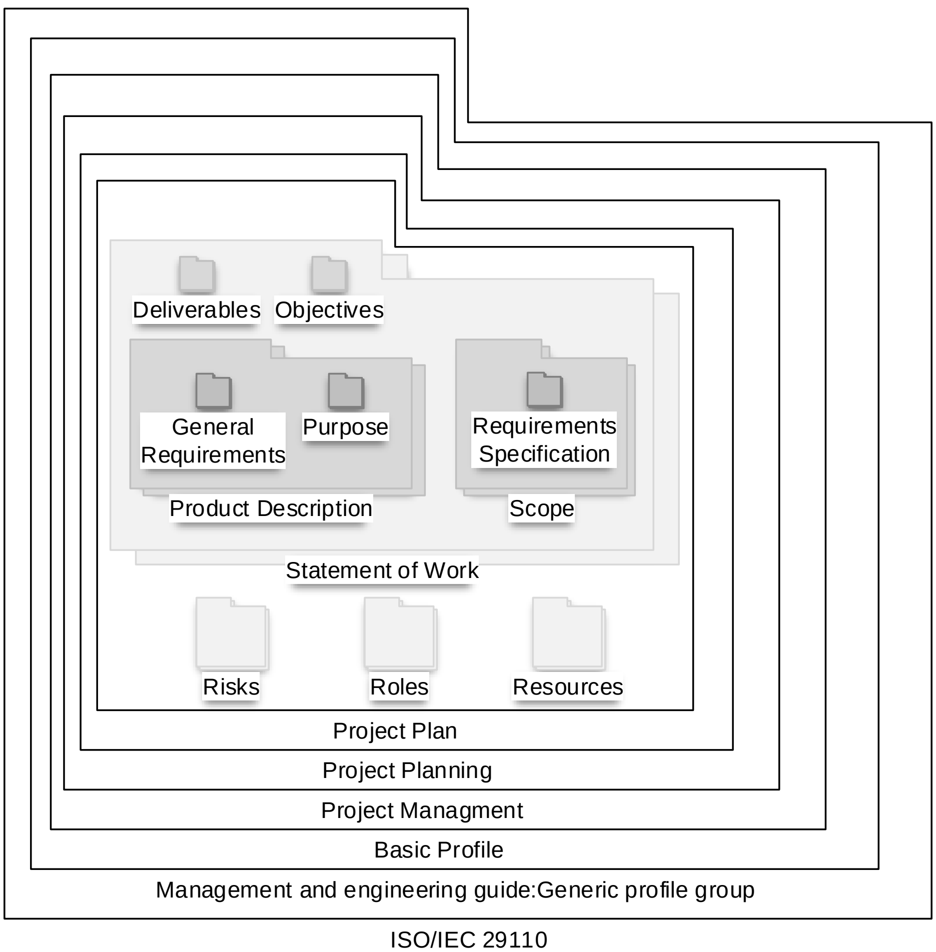
4 A Pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110

In this M.Sc. Thesis we formulate a method for real-life VSE software process implementations. The proposed solution is based on work products related to requirements elicitation in the project management and software implementation processes defined in the basic profile of the ISO/IEC 29110 standard. The proposed approach prioritizes the accurate requirements elicitation by using pre-conceptual schemas. By implementing such a method, we propose an alternative to VSEs when implementing the basic profile of the ISO/IEC 29110 in early stages.

4.1 Pre-conceptual schemas for representing requirements elicitation in the ISO/IEC 29110 domain

Pre-conceptual schemas reduce the gap between stakeholder and analyst. Such schemas are aimed to ease communication by representing a domain in a graphical environment. We construct a guide to be followed when implementing the basic profile in early stages by using pre-conceptual schemas. The method proposed in this M.Sc. Thesis includes five main concepts which are defined in the basic profile. Four concepts (roles, statement of work, resources, and risks) are described in the project planning activity, which is the first activity in the project management process. Likewise, one concept (requirements specification) is detailed in the software requirements analysis activity, which is the second activity in the software implementation process. Such concepts are intended to be main components of the requirements elicitation procedure represented in our pre-conceptual schema. The requirements specification work product is defined as part of the scope in the statement of work. Therefore, specification requirement tasks are performed in early stages (before project plan completion). Such a fact allows technical leader, analyst, and development team for verifing and validating the requirements already elicited when developing the software implementation process. Requirements in the software implementation process can be added by elaborating a change request document. Early specification of requirements allows for constructing of a solid project plan, reducing the possibilities of changes in middle and final stages of the project, and avoiding over costs. As shown in Figure 4-1, all work products included are part of the macro work product Project Plan.

Figure 4-1: Work products used in the ISO/IEC 29110 domain (The Authors based on ISO/IEC, 2011).

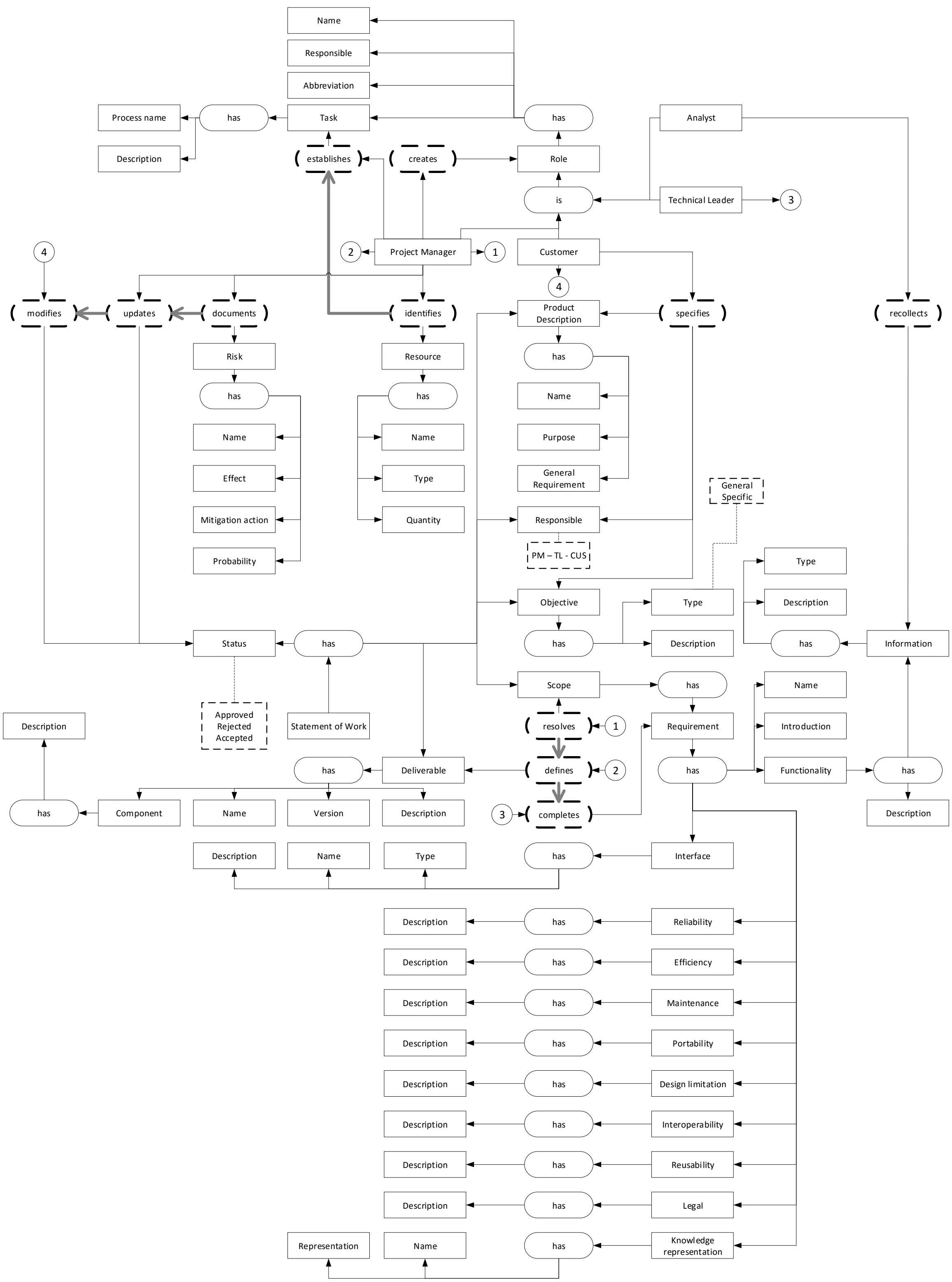


According to the ISO/IEC (2011), developing a work product is a procedure which includes: an input work product, a task to be performed, a responsible for developing such task, and the resulting work product. Detailed specifications about the work products defined in this approach are shown in Table 4-1.

Table 4-1: Role responsible, task list, input work products, and output work products in requirements elicitation (The Authors based on ISO/IEC, 2011).

Role responsible	Task List	Input work products	Output work product
PM–TL–CUS	– Specify product description, purpose, general requirements, responsible, and objectives. – Resolve the scope of the project. – Define the deliverables. – Update the project status. – Approve the statement of work.		Statement of work
PM-TL-AN	– Resolve the requirements name, introduction, and the functionality description, information, and type. – Complete the corresponding data of the following requirements aspects: • Interface • Reliability • Efficiency • Maintenance • Portability • Design limitations • Interoperability • Reusability • Legal • Knowledge representation	Statement of work	Requirements specification
PM	– Identify the resources: human, material, equipment, tools, standards, etc.	Statement of work	Project Plan -Resources
PM	– Create roles involved in the project. – Establish the tasks to be performed for producing the deliverables included in the statement of work. – Include tasks in the SI process along with verification, validation, and reviews with the customer and work team to assure quality. – Identify the delivery instructions tasks.	Statement of work	Project Plan -Roles
PM	– Document the risks which may affect the project.	Statement of work	Project Plan -Risks

Figure 4-2: Pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110 (The Authors).



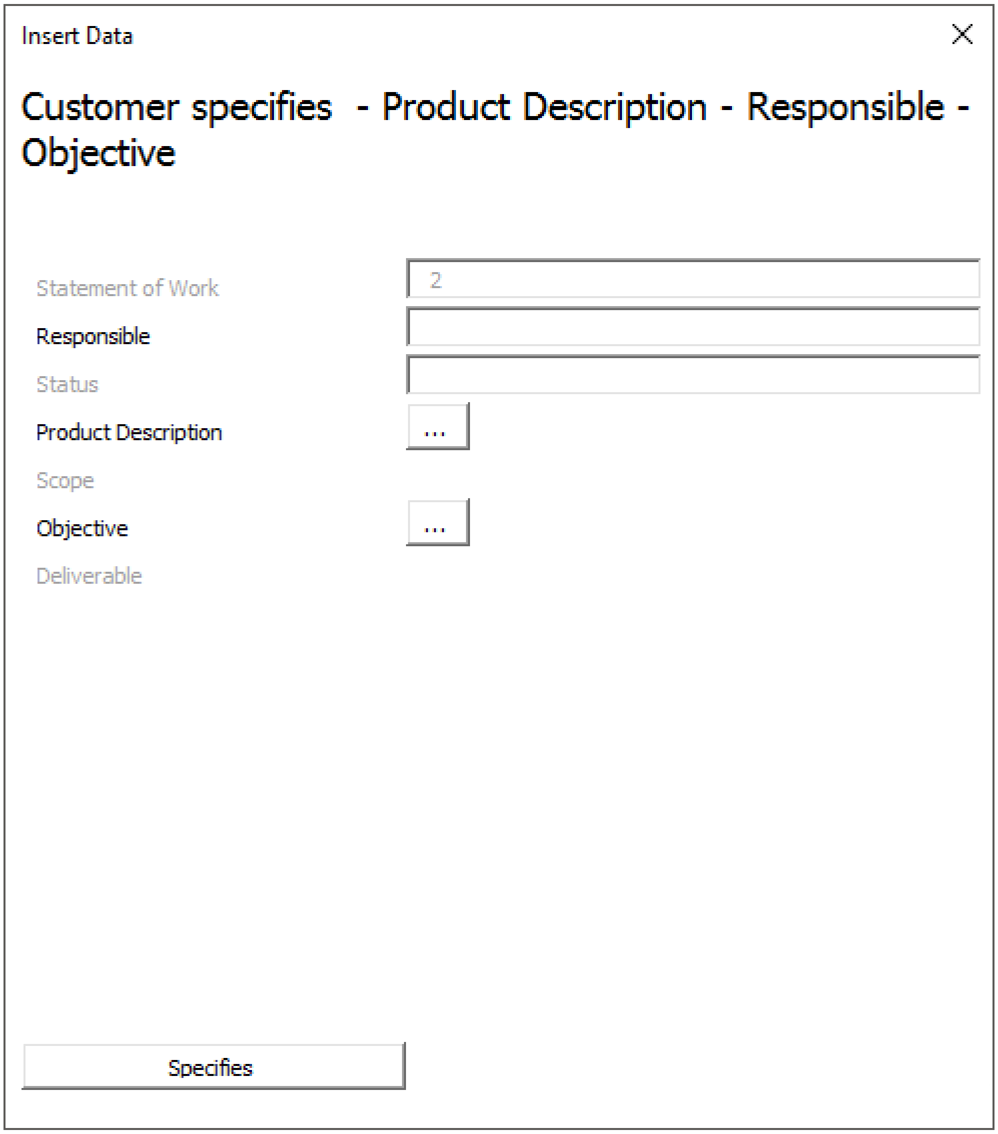
In Figure 4-2 we propose the pre-conceptual schema constructed for representing the requirements elicitation procedure. This schema includes the previously specified five main concepts. The tasks regarding such work products should be performed in a sequential order. The first task is the creation of roles (name, responsible, and abbreviation). The project manager defines the basic roles necessary for the project management process according to the ISO/IEC 29110. Next, the customer specifies the product description (name, purpose, and general requirement) and the objectives (general and specific), which are the first work products of the statement of work. The responsible of the statement of work are the project manager, the technical leader, and the client. Consequently, the project manager together with the customer resolves the scope of the project by defining the requirements (name and introduction) based on their functionality. The analyst completes the functionality requirements by collecting additional information (type and description) from the customer. Next, the project manager defines the deliverables (name, description, version, and component) obtained from the functional requirements previously established. Finally, the technical leader completes the statement of work by fulfilling the interface details (description, name, and type), constructing the knowledge representations of the project (pre-conceptual schemas, class diagrams, communication diagrams, entity-relationship models, state machine diagrams, use case diagrams, etc.), and specifying the remaining requirements such as reliability, efficiency, maintenance, portability, design limitations, interoperability, reusability, and legals of the project.

After this procedure, both the statement of work and the requirements specification are complete. Consequently, the project manager has an absolute conception of the project to be carried out and the implications involved. Such understanding of the project gives the project manager capability to identify the resources needed to carry out the development (human, equipment, state, supplements, furniture, hardware, digital, etc.) and classify them by type, and quantity. In addition, the project manager establishes the tasks to be performed by the roles previously defined in the statement of work and corroborated in the resource definition. Besides, the project manager documents and identifies possible risk occurrences during the project by including the effect of such risks, mitigation actions, and probabilities. Finally, by establishing the statement of work, the requirements specification, the roles involved, the resources needed, and the possible risk occurrences, the project manager updates the statement of work status by accepting/rejecting the project. If the statement of work is accepted, the project manager meets the customer, which modifies the statement of work status by approving/rejecting the project. The customer acceptance is the final part of the requirements elicitation method proposed. In case of stablishing a “rejection” in the statement of work status (in both cases, when the project manager updates the status and when the customer modifies the status), the requirements elicitation should be performed again in order to define feasible requirements of the project to be developed.

The method proposed in this M.Sc. Thesis provides VSEs with an accurate and unambiguous understanding of the project to be developed. Likewise, with the identification of roles, resources, and risks involved, VSEs have the ability to define the viability of such a project. By reaching an agreement between the customer and the organization about the project domain, negotiations about cost and delivery time can be defined along with the rest of the project plan.

The pre-conceptual schema developed in this M.Sc. Thesis includes an executable interface for interactive purposes. By using a cognitive graphical interface, the project manager can systematically elicit the requirements by following the pre-conceptual schema. Project manager, technical leader, and analyst can fulfill the data of each concept in order to complete the work products needed at this stage. Such an interactive alternative is aimed to ease the process by providing a map to be followed for accomplishing an agile and lightweight requirements elicitation process. Besides, the interactive representation of the project domain improves the communication between the customer and the project manager, enhancing the probability of an agreement among the stakeholders and the analyst. The executable interface is developed in a Microsoft Visio™ file (*.vsdm), and includes an excel™ object which is composed of the schema structure, and is used to save the data of each concept. Likewise, the program is developed in Visual Basic for Applications™. In Figure 4-3, we show an example of the executable pre-conceptual schema interface when clicking in the dynamic relationship “specifies”.

Figure 4-3: Example of an executable pre-conceptual schema interface (The Authors).



As illustrated in Figure 4-3, the interface restricts the customer only to specify the responsible, product description, and objective in the statement of work, according to the pre-conceptual schema structure. In Table 4-2, we show part of the data structure stored in the excel™ sheet when compiling the code.

Table 4-2: Part of a data structure stored in the excel™ sheet (The Authors).

Concepts	Notes	Static relationships	Dynamic relationships	References
Role	PM–TL-CUS	is	updates	2 (Project Manager)
Project Manager	Approved Rejected Accepted	has	specifies	1 (Project Manager)
Customer		has	resolves	1 (Project Manager)
Statement of Work		has	defines	2 (Project Manager)
Product Description		has	documents	2 (Project Manager)
Responsible (Leaf)		has	creates	3 (Technical Leader)
Purpose (Leaf)		has	identifies	3 (Technical Leader)
General Requirement (Leaf)		has	establishes	4 (Customer)
Scope		has	approves	4 (Customer)
Requirement		has	recollects	
Resource		has	completes	
Type (Leaf)		has		
Quantity (Leaf)		has		
Objective		has		
Deliverable		has		
Description (Leaf)		has		
Component		has		
Description (Leaf)		has		
Abbreviation (Leaf)		has		
Knowledge representation		has		
Functionality		has		
Task		has		
Responsible (Leaf)		has		
Description (Leaf)		has		
Process name (Leaf)				
Technical Leader				
Name (Leaf)				
Version (Leaf)				
Description (Leaf)				
Name (Leaf)				
Risk				
Name (Leaf)				
Effect (Leaf)				
Mitigation actions (Leaf)				
Probability (Leaf)				

5 Case study

5.1 Requirements elicitation for a mobile medical recommendation model

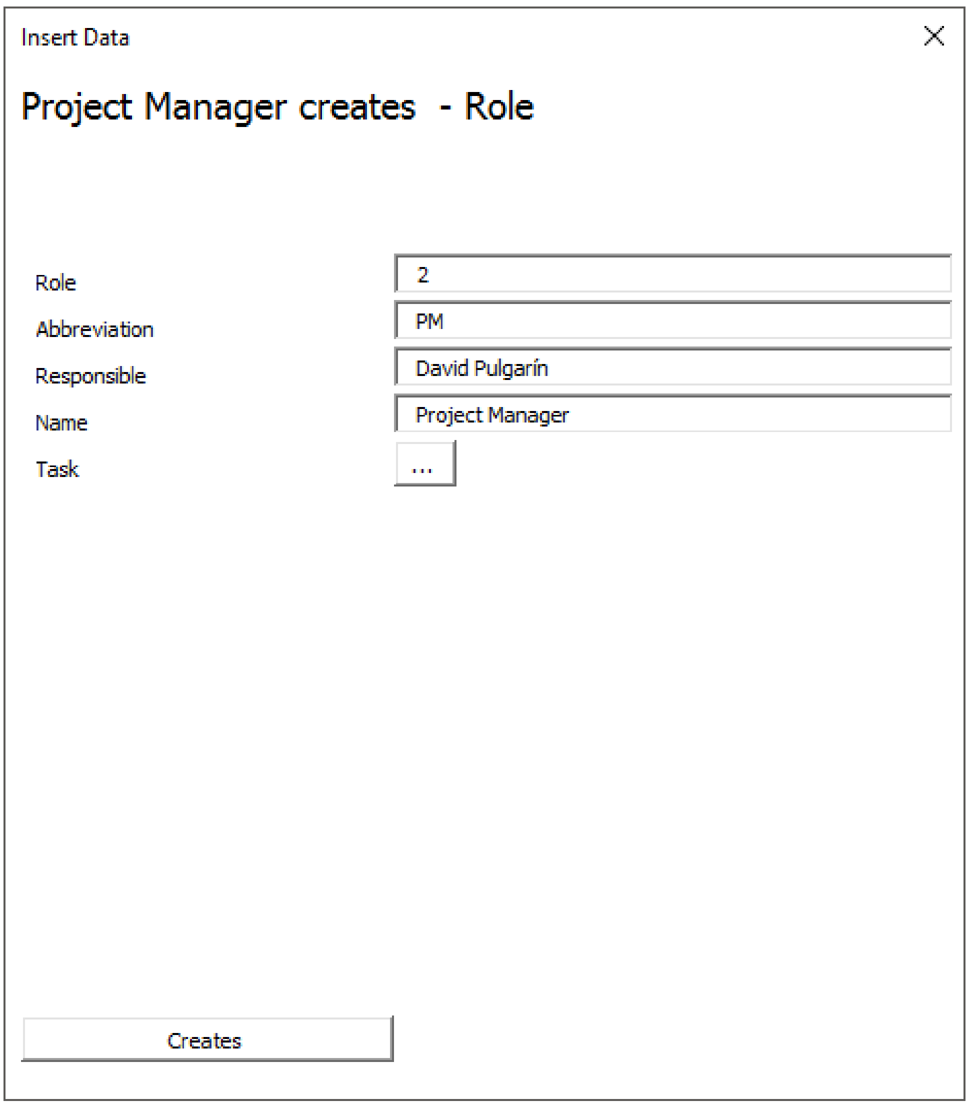
COLCIENCIAS is a government entity responsible for promoting public policies to improve science, technology, and innovation in Colombia. Young researches across the country annually present their projects in order to be financed by such an entity. The case study we propose in this M.Sc. Thesis is a research project approved and funded in April 2018 by the annual program established by COLCIENCIAS named *jóvenes Investigadores* (young researches). Such a project is developed in order to create a mobile medical recommendation model (by using drones) for improving measures, monitoring and controlling sensitive diseases related to weather in the metropolitan area of Medellin. By implementing the basic profile included in the ISO/IEC 29110 and using pre-conceptual schemas, we elicit the requirements explained by the stakeholder. Such a process is represented in the graphical-interactive schema, which allows both stakeholder and analyst to embrace the solution in a language understandable by each other. The executable pre-conceptual schema is implemented along the complete requirements elicitation procedure. The requirements elicitation is part of the project plan included in the project management process developed for the stakeholder project.

The first step for starting up the requirements elicitation is the schedule of an initial meeting with the customer in order to specify the first work products of the statement of work. By following the pre-conceptual schema, we establish responsible, roles, and role abbreviations (see Table 5-1; Figure 5-1).

Table 5-1: Roles Nomenclature (The Authors).

Role	Abbreviation	Name
Client	CUS	COLCIENCIAS
Project Manager	PM	David Pulgarín
Technical Leader - Analyst	TL	Jaime Alberto Guzmán Luna

Figure 5-1: Creation of a role in the executable pre-conceptual schema (The Authors).



Next, we specify the product description, purpose, general requirements, responsible, general objective, and specific objectives (see Table 5-2; Figure 5-2).

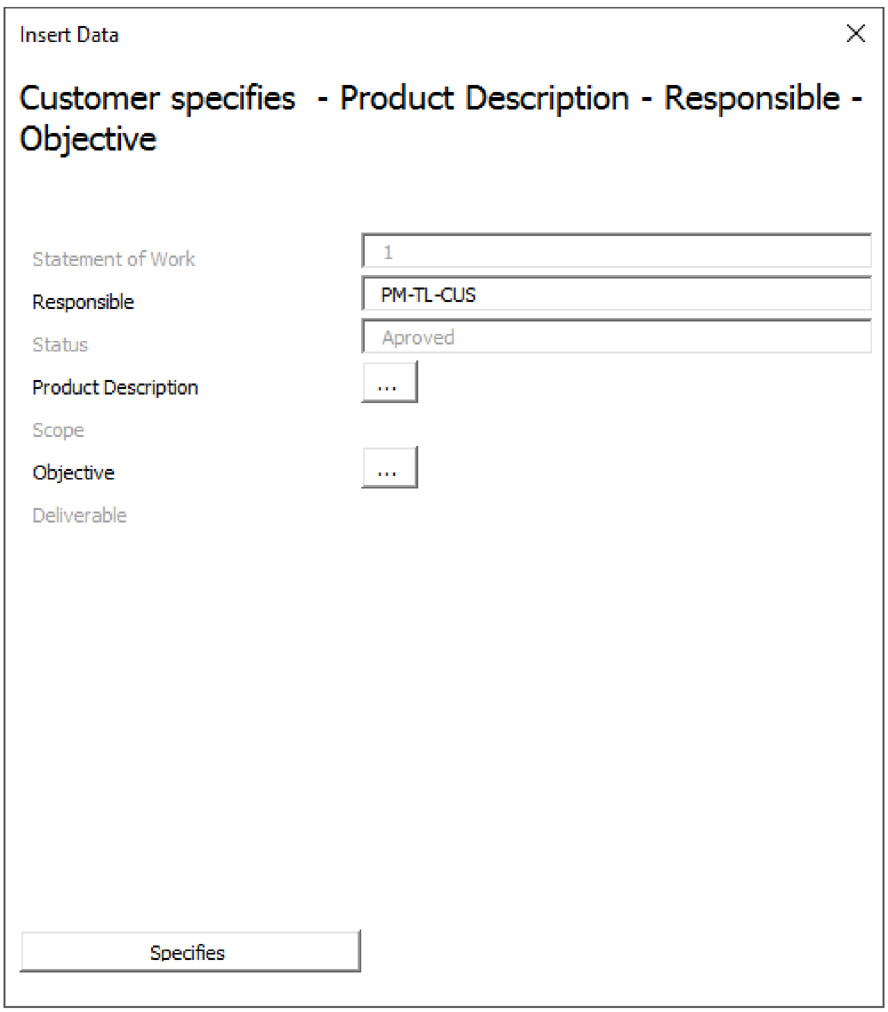
Table 5-2: Product description, responsible, and objectives of the statement of work (1 out of 2; The Authors).

Product description	**Name** ▪ Mobile medical recommendation model, with the use of drones, for improving measures, monitoring, and controlling climate-sensitive diseases related to the weather in the metropolitan area of Medellin.
	Purpose ▪ Measurement of climatic variables in certain geographical areas within the city of Medellín to identify possible diseases in the environment.
	General Requirements ▪ Find new solutions by researching viable new alternatives in different fields of science.
Responsible	▪ Project manager (PM) ▪ Technical leader (TL) ▪ Client (CUS).
Objectives	**General** ▪ Develop a medical recommendation model by using a mobile application, with the use of drones, allowing the user for knowing the climate-sensitive diseases in a certain locations of the metropolitan area of Medellín, and based on such a fact, minimizing the impact on health.
	Specific ▪ Characterize the most important climate-sensitive diseases, which will be tracked by the drone sensors, considering the articulation to a specific mapping of the most common diseases, defined by the medical area. ▪ Establish a well-supported course, which relates the main sources of climate-sensitive diseases in the environment and the recommendation model of the mobile application to define the parameters, which will be presented to the user. ▪ Design an architectural model, which allows for articulating climate sampling by using drones, with the identification of several climate-sensitive diseases captured by the sensors and the mobile medical recommendation system.

Table 5-2: Product description, responsible, and objectives of the statement of work (2 out of 2; The Authors).

Objectives	**Specific** ▪ Implement the proposed model with special care to communication protocols, so an appropriate communication system for the project could be chosen. ▪ Evaluate the behavior of the model in a real environment and its adequate functionality in presenting the user a mobile device with medical recommendations, according to weather-sensitive diseases in the environment to define the strengths and weaknesses of the model.

Figure 5-2: Specification of product description, responsible, and objective in the executable pre-conceptual schema (The Authors).



In addition, we resolve the first work products of the project scope according to the functional requirements by including the collection of additional information regarding the project (see Table 5-3; Figure 5-3).

Table 5-3: Project scope (name, introduction, and functionality) included in the statement of work (1 out of 2; The Authors).

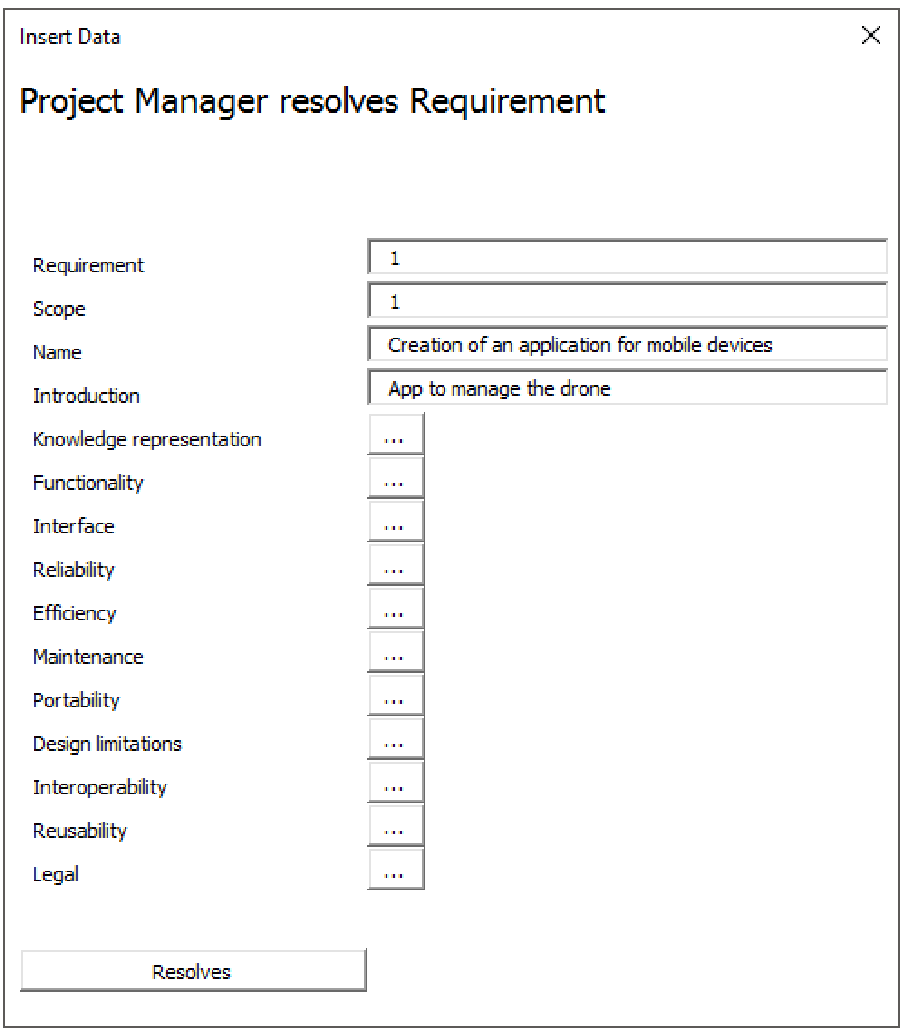
Requirement	Information
Creation of an application for mobile devices	**Introduction** ▪ Application to manage the drone.

	Functionality
	▪ Download the application from Google Play Store ™.
	  ▪ Information type: digital.
	  ▪ Information description: only downloadable from Android.
	▪ Registration of new users in the application by using a data entry form.
	▪ Login of registered users in the application.
	▪ Georeferenced detection of the user location.
	▪ Map displaying the flights whose trajectory was closest to the user current location.
	  ▪ Information type: design.
	  ▪ Information description: information should be visualized in a 1000-meter radio.
	▪ Visualization of climate variables referring to each zone overflown: temperature, moisture, precipitation, and solar radiation.
	▪ Glossary of parameterized colors for each variable based on their level of criticality.
	  ▪ Information Type: Construction.
	  ▪ Information Description: the project manager in the database should previously insert the Information.
	▪ Visualization of possible diseases to arise in each evaluated area.
	▪ Recommendations to prevent possible diseases detected in each evaluated area.

Table 5-3: Project scope (name, introduction, and functionality) included in the statement of work (2 out of 2; The Authors).

Requirement	Information
Commercial drone	**Introduction** ▪ 200-meter reach needed for the drone.
	Functionality ▪ A sensor to measure solar radiation. ▪ A sensor to measure temperature, humidity, and precipitation. ▪ Two XBee communication platforms with a range between 100m and 1000m used to transmit the data obtained by the sensors.

Figure 5-3: Resolution of requirements in the executable pre-conceptual schema (The Authors).

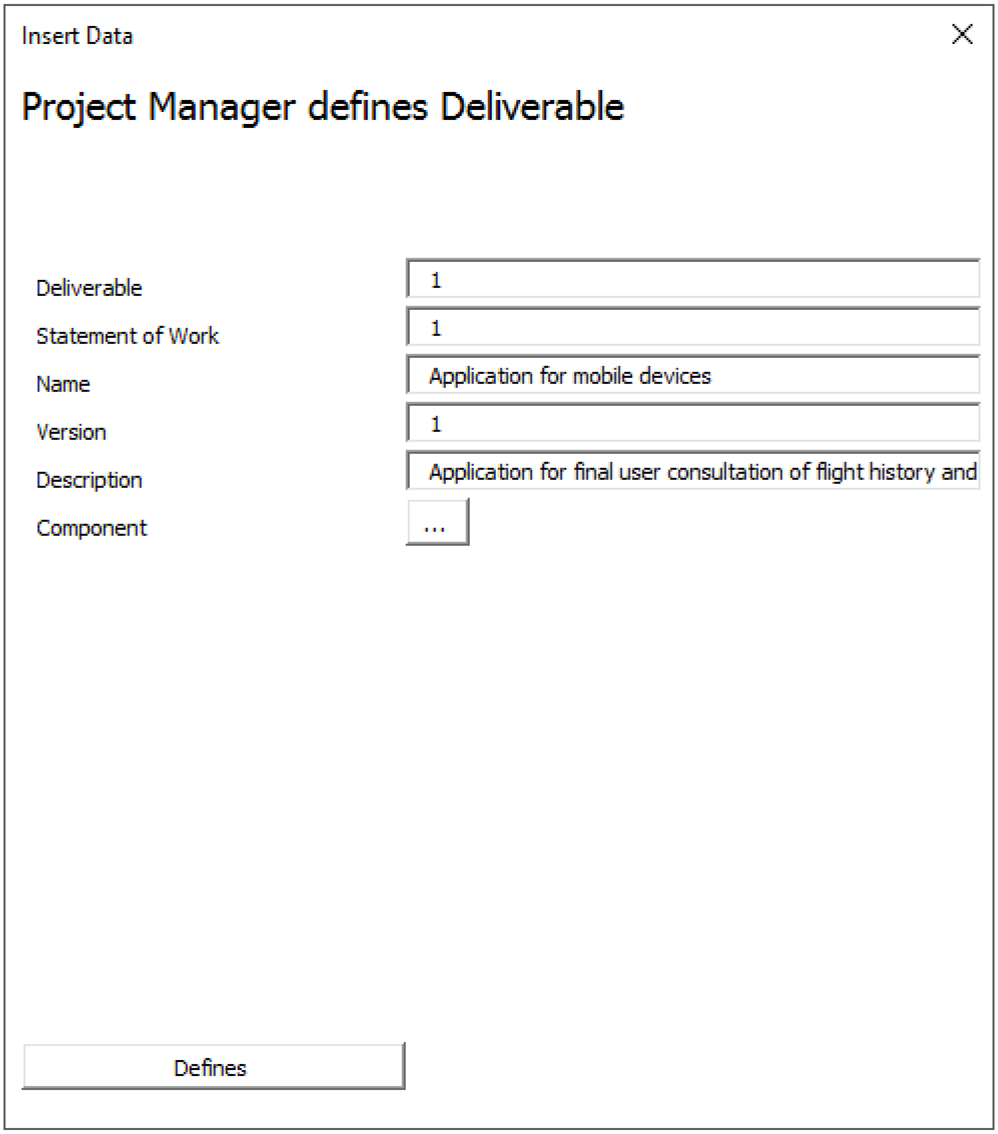


Finally, the initial meeting ends with the definition of the deliverables (see Figure 5-4), including the name, description, and a list of the components of each product (see Table 5-4).

Table 5-4: Deliverables included in the statement of work (The Authors).

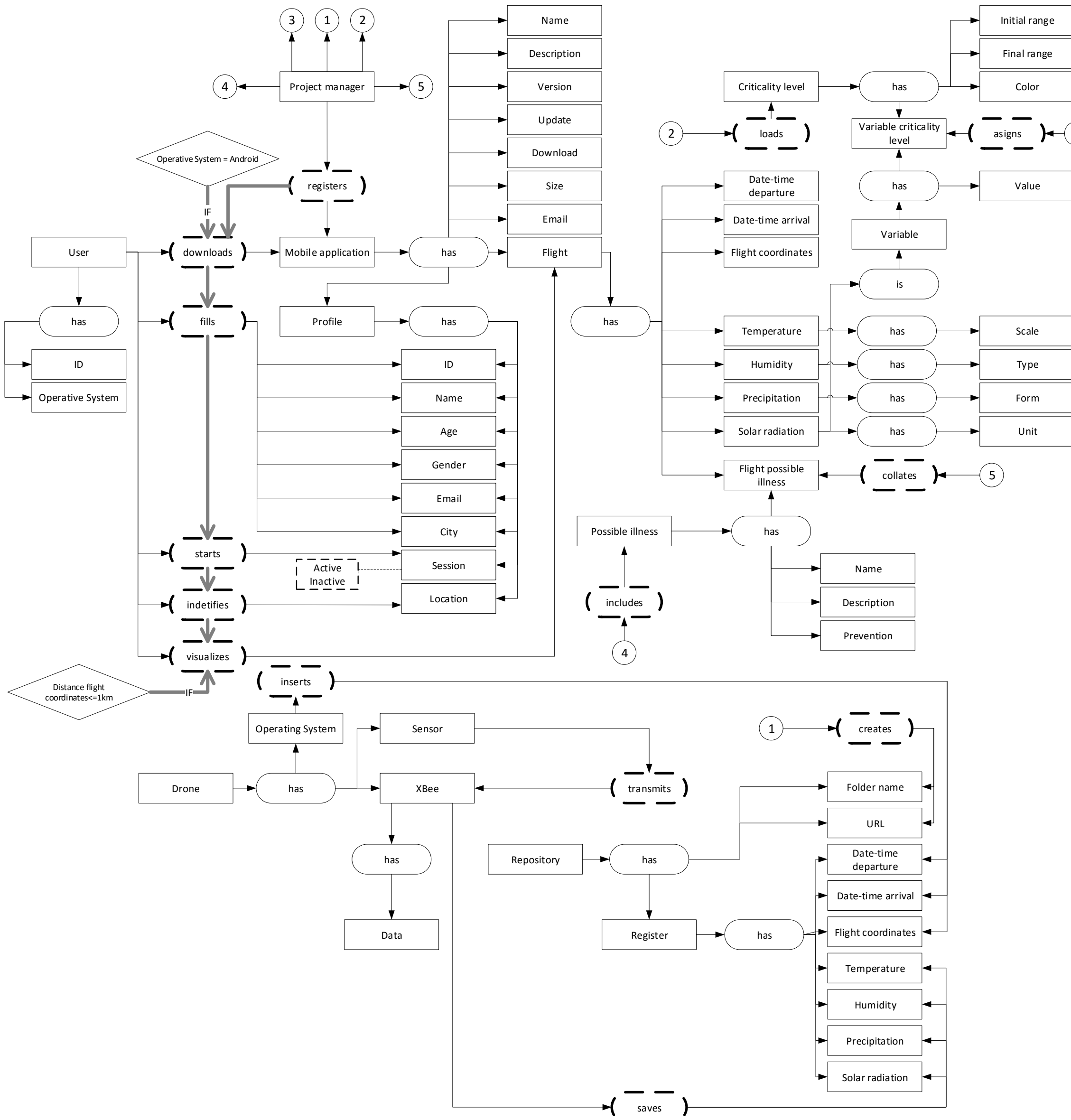
Deliverable	Information
Application for mobile devices	**Description** ▪ Application for final user review of flight history and information.
	Version ▪ 1.0.
	Components ▪ Ability to download from the Google Play Store. ▪ Registration of new users. ▪ Login of registered users. ▪ Automatic detection of the user location. ▪ Display and selection of flight history near to current location. ▪ Visualization of four climatic variables referring to each overflowed area: temperature, humidity, precipitation, and solar radiation. ▪ Fan of defined colors for each variable based on their level of criticality. ▪ Detection of possible diseases to be raised together with the respective recommendations for their prevention of each overflow zone.
Modified commercial drone	**Description** ▪ Commercial drone with modifications for the detection and remote broadcast of variables measured in flight.
	Version ▪ 1.0.
	Components ▪ A sensor to measure solar radiation. ▪ A sensor to measure temperature, humidity, and precipitation. ▪ Two XBee communication platforms with a range between 100m and 1000m used to transmit the data obtained by the sensors. ▪ An information receiver, which stores the data received on a local hard disk. ▪ Use of an indigenous application of the commercial drone for its piloting and management.

Figure 5-4: Definition of deliverables in the executable pre-conceptual schema (The Authors).



In the next meeting with the customer, we evaluate the project domain in order to reach an agreement in terms of functionality. In the case study, we verify the project along with the customer by using a pre-conceptual schema in order to obtain an accurate and wide view of the domain (see Figure 5-5).

Figure 5-5: Project pre-conceptual schema (The Authors).



Next, we complete the requirements specification by defining information regarding the interface, reliability, efficiency, maintenance, portability, design limitations, interoperability, reusability, and legal aspects (see Figure 5-3; Table 5-5).

Table 5-5: Project scope (name, interface, reliability, efficiency, maintenance, portability, design limitations, interoperability, reusability, and legal) included in the statement of work (1 out of 2; The Authors).

Name ▪ Creation of an application for mobile devices.

Interface
▪ Register app in Google Play Store.
    ▪ Description: interface design for administrative purposes only.
    ▪ Type: internal.
▪ Login screen.
    ▪ Description: two textboxes for username and password, and a two-command button for login or exit.
    ▪ Type: external.
▪ Visualization map.
    ▪ Description: show actual location, previous areas covered by the drone, possible diseases and methods for preventing such diseases.
    ▪ Type: external.

Reliability ▪ Information is stored in Google Drive (cloud). ▪ A physical disk is stored in a safe environment; such a disk is updated once every month.
Efficiency ▪ Software size is 4 Mb with no extra downloads necessary. ▪ The app is playable in a device with GPS incorporated. ▪ The app is playable in a device with gyroscope incorporated. ▪ Transactions for insert, update, or delete are excluded. ▪ Select procedures are included.
Maintenance ▪ Software development should be documented in every module, class, and procedure. ▪ Every variable should be defined in lower case. ▪ Every procedure or function should be defined in upper case. ▪ Variables and functions should be mnemonic. ▪ Classes should be stored in different files. ▪ Software should be parameterized for scalability purposes.

Table 5-5: Project scope (name, interface, reliability, efficiency, maintenance, portability, design limitations, interoperability, reusability, and legal) included in the statement of work (2 out of 2; The Authors).

Portability ▪ The software could be downloaded in PCs, cell phones, tablets, laptops, and any device with an Android operating system.
Design limitations ▪ The app will include a range for visualizing the history of flights in a radio of 1000 meters. ▪ The project manager previously defines critical colors for variables.
Interoperability ▪ No interoperability is applied.
Reusability ▪ No reusability is defined at this software version.

We define a set of conceptual diagrams as knowledge representations included in the scope. By using a pre-conceptual schema, we model the domain and automatically obtain some UML diagrams (Zapata, Gelbukh & Arango, 2006a; Zapata, Gelbukh & Arango, 2006b) such as class, communication, and state machine. Likewise, we obtain the use case diagram (Zapata, Villa & Chaverra, 2014) and the entity-relationship model (Chaverra, 2011) with the pre-conceptual schema constructed.

- UML class diagram (see Figure 5-6).
- UML communication diagram (see Figure 5-7).
- UML state machine diagram (see Figure 5-8).
- Use case diagram (see Figure 5-9; Table 5-6 to Table 5-19).
    - Use case Create Repository.
    - Use case Insert Record.
    - Use case Save variable.
    - Use case Data transmit.
    - Use case Register mobile application.
    - Use case Load criticality level.
    - Use case Include possible illness.
    - Use case Assign variable criticality level.
    - Use case Collate flight possible illness.
    - Use case Download mobile application.

- Use case Enter profile.
- Use case Log in.
- Use case Find location.
- Use case View flight.

- Entity–relationship model (see Figure 5-10)

Figure 5-6: UML class diagram (The Authors).

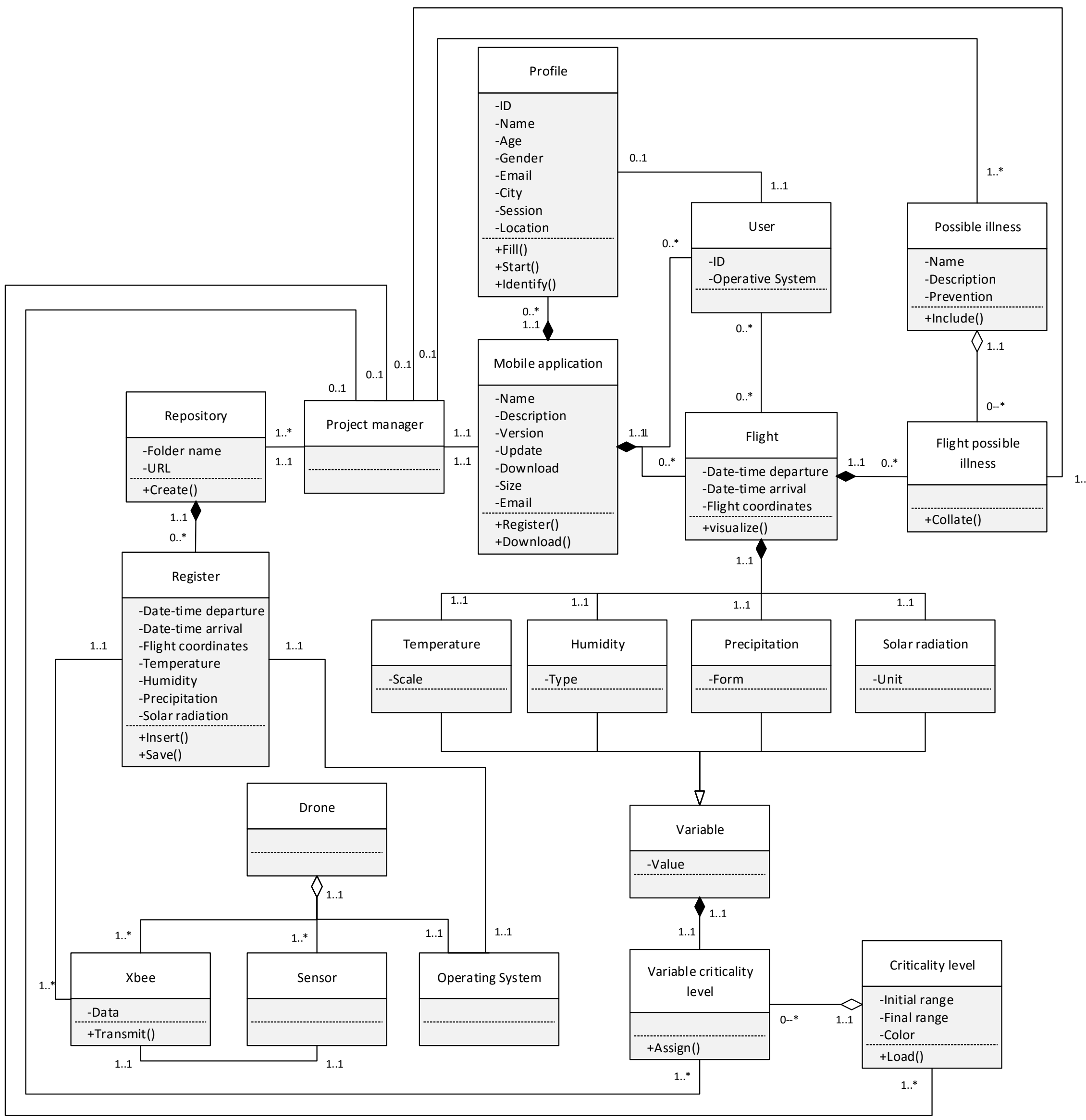


Figure 5-7: UML communication diagram (The Authors).

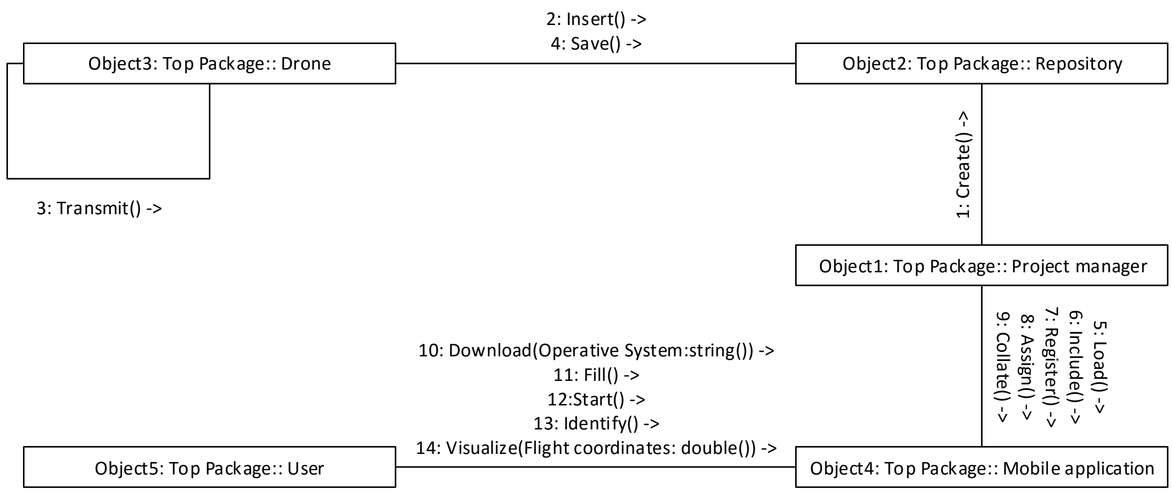


Figure 5-8: UML state machine diagram (The Authors).

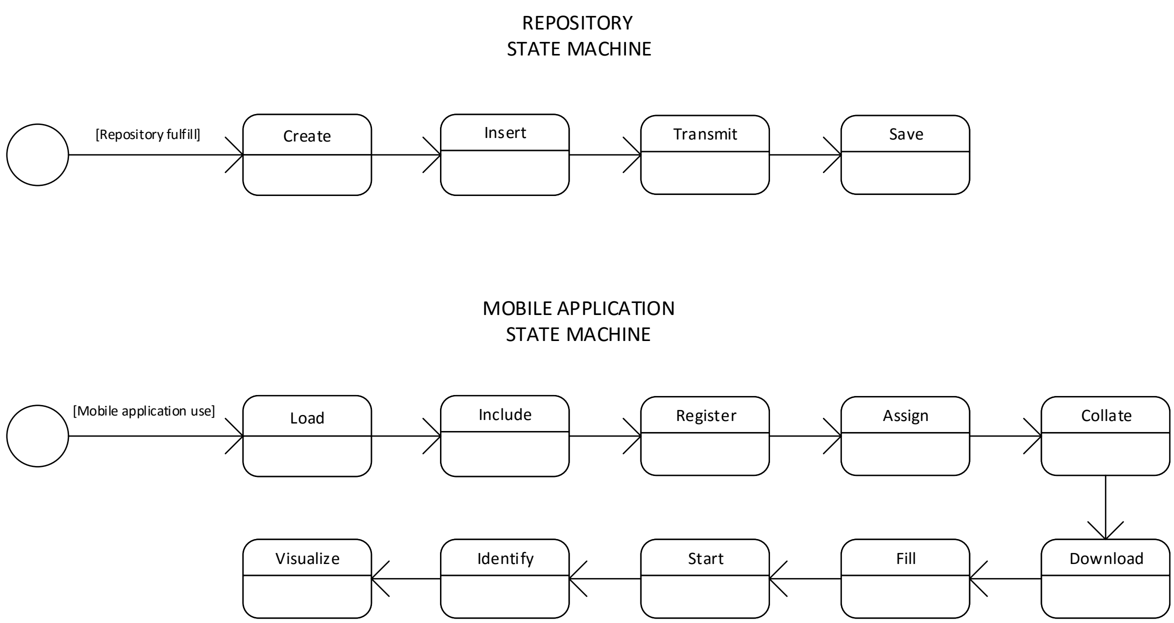


Figure 5-9: Use case diagram (The Authors).

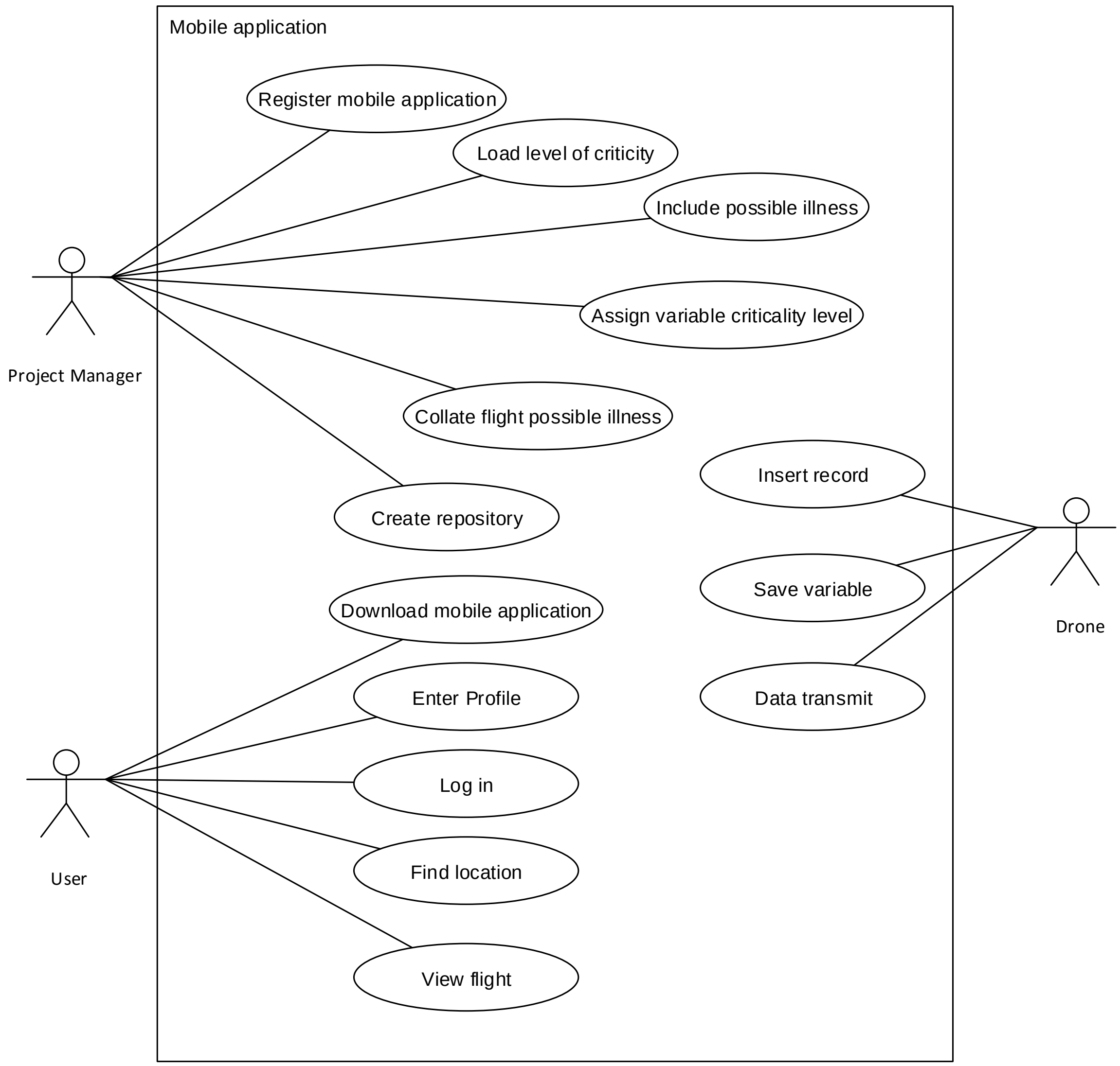


Table 5-6: Use case Create Repository (The Authors).

Use case	Create repository
Actors	Project manager
Summary	The project manager creates the repository by selecting the folder name and the URL and defines disk space for recording date-time entry, date-time departure, flight coordinates, temperature, humidity, precipitation, and solar radiation.
Pre-conditions	No repository created.
Post-conditions	A repository to store flight information.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The project manager creates a repository	Stores usage variables for flight.

Table 5-7: Use case Insert Record (The Authors).

Use case	Insert record
Actors	Drone
Summary	The drone sensors register in the repository the arrived date-time, departure date-time, and flight coordinates.
Pre-conditions	Repository created.
Post-conditions	Initial flight registration.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
Drone issues data and registers it in the repository.	Stores usage variables for flight.

Table 5-8: Use case Save variable (The Authors).

Use case	Save variable
Actors	Drone
Summary	By using sensors, the drone receives temperature, humidity, precipitation, and solar radiation information. It sends the data from the Xbee microcontroller and saves the variables in the repository.
Pre-conditions	Repository created.
Post-conditions	Record of flight variables
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
Drone saves variables data in the repository.	Stores usage variables for flight.

Table 5-9: Use case Data transmit (The Authors).

Use case	Data transmit
Actors	Drone
Summary	Transmits data from the sensors incorporated in the drone to the Xbee microcontroller.
Pre-conditions	Installation of sensors and microcontroller
Post-conditions	Sending raw data to the microcontroller
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
Drone sensor sends data to Xbee.	Receives information in a schematic way.

Table 5-10: Use case Register mobile application (The Authors).

Use case	Register mobile application
Actors	Project manager
Summary	The project manager registers the mobile application by specifying the name, description, version, update, size, and mail.
Pre-conditions	Application developed
Post-conditions	Application registered in Google play
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
Project manager register application in Google Play.	Active status of the application for user download.

Table 5-11: Use case Load criticality level (The Authors).

Use case	Load criticality level
Actors	Project manager
Summary	The project manager loads the criticality levels for each variable.
Pre-conditions	Application developed
Post-conditions	Criticality level table complete in database.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
Project manager loads the criticality levels.	Criticality level table from database fulfill

Table 5-12: Use case Include possible illness (The Authors).

Use case	Include possible illness
Actors	Project manager
Summary	The project manager Includes possible illness regarding the area.
Pre-conditions	Application developed
Post-conditions	Possible illness table complete in database.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The PM includes possible illness.	Possible illness table from database fulfill

Table 5-13: Use case Assign variable criticality level (The Authors).

Use case	Assign variable criticality level
Actors	Project manager
Summary	The project manager assigns the equivalence between the variables and the criticality levels for each flight.
Pre-conditions	Application developed Load criticality level Register mobile application
Post-conditions	Variable criticality level table complete in database.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The project manager assigns the variables and the criticality levels for each flight.	Variable criticality level table from database fulfill

Table 5-14: Use case Collate flight possible illness (The Authors).

Use case	Collate flight possible illness
Actors	Project manager
Summary	The project manager assigns the equivalence between the flight and the possible illnesses for each flight.
Pre-conditions	Application developed Include possible illness Register mobile application
Post-conditions	Flight possible illness table complete in the database.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The project manager creates equivalence between the flight and the possible illnesses.	Flight possible illness table from database fulfill

Table 5-15: Use case Download mobile application (The Authors).

Use case	Download mobile application
Actors	User
Summary	User downloads mobile application from Google Play.
Pre-conditions	The Android operating system installed.
Post-conditions	Application installed on mobile device.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
User downloads the mobile application.	Increase the number of user download records by one.

Table 5-16: Use case Enter profile (The Authors).

Use case	Enter profile
Actors	User
Summary	The user enters their profile data including ID, name, age, gender, mail, and city.
Pre-conditions	Download application from Google Play.
Post-conditions	Registered user profile
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The user enters profile data.	Registers data entered by the user in the database.

Table 5-17: Use case Log in (The Authors).

Use case	Log in
Actors	User
Summary	The user starts the session by using the application.
Pre-conditions	Registered user profile
Post-conditions	Active user session.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
The user starts a session in the application	Changes the session state of the user from inactive to active.

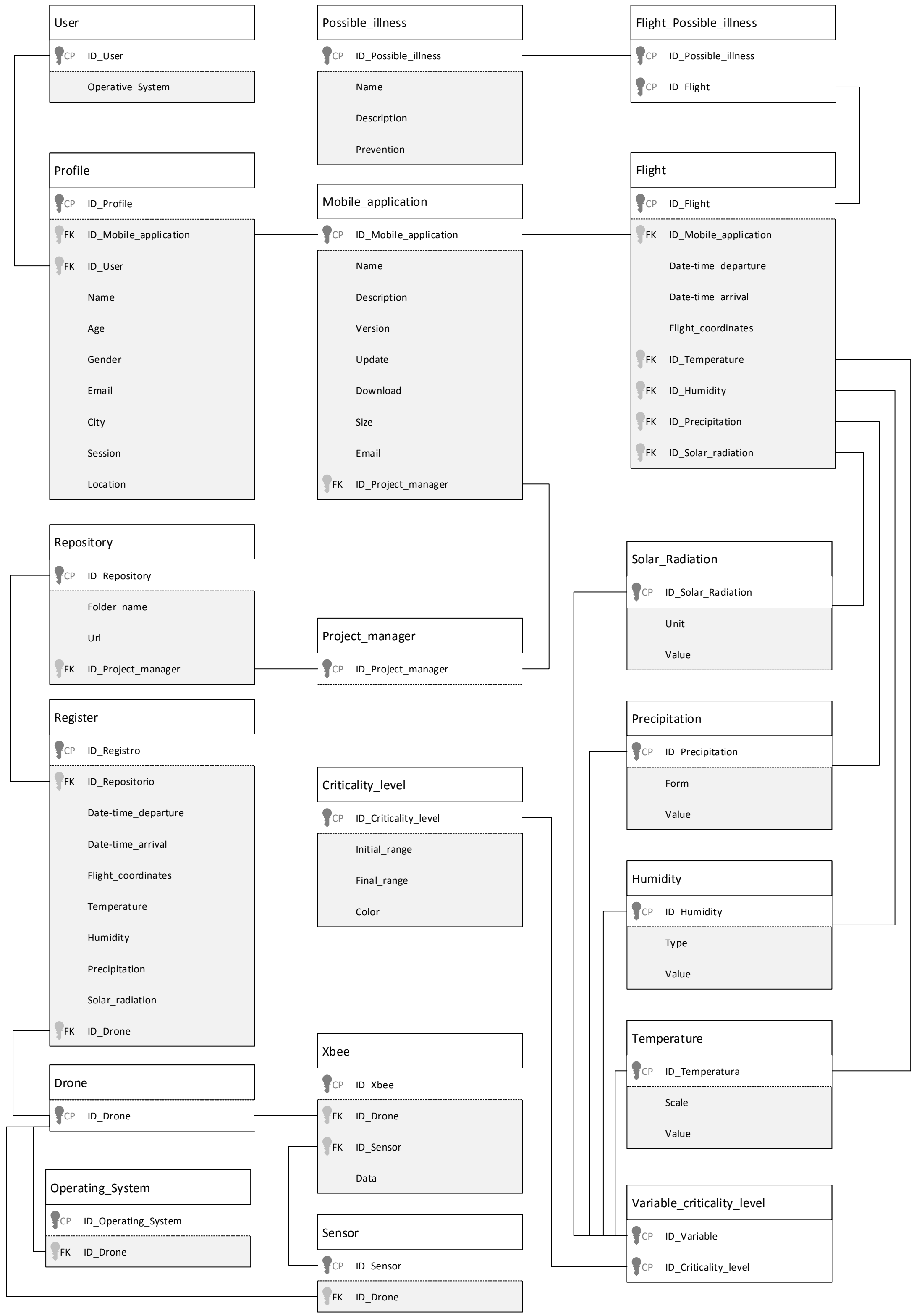
Table 5-18: Use case Find location (The Authors).

Use case	Find location
Actors	User
Summary	The user notices the location in real time.
Pre-conditions	Active user session.
Post-conditions	The map displayed with the user current location.
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
User displays location on the mobile device.	Shows the georeferenced location of the application user.

Table 5-19: Use case View flight (The Authors).

Use case	View flight
Actors	User
Summary	The device displays flight data including departure date-time, arrival date-time, flight coordinates, temperature, humidity, precipitation, solar radiation, possible diseases, criticality level, and prevention of possible diseases.
Pre-conditions	The map displayed with the user current location.
Post-conditions	Visualization of all the variables and application data
Includes	
Extends	
Inherit from	
Event flow	
Actor	**System**
User visualizes areas overflown by the drone and its variables.	Display information segmented by flights close to the user location.

Figure 5-10: Entity–relationship model (The Authors).



After completing the statement of work, we identify the resources needed to complete the project by including the type and quantity values (see Table 5-20; Figure 5-11)

Table 5-20: Resources used in the project (The Authors).

Resource	**Name**	**Quantity**
Human	Project manager Technical leader – Analyst	1 1
Equipment	Desktop Modem	1 1
Estate	Office	1
Supplements	Sheet Folder	10 5
Furniture	Desk Chair	1 1
Hardware	Drone Cell phone Sensor Microcontroller	1 1 2 2
Digital	Receiving application Drone management application SQL Server 2017 Visual studio 2015 - C# Ajax toolkit for visual studio	1 1 1 1 1

Figure 5-11: Update screen of resources in the executable pre-conceptual schema (The Authors).

Seleccione Registro

Resource	Type	Quantity	Name
1	Human	1	Project manager
2	Human	1	Technical leader
3	Human	1	Analyst
4	Equipment	1	Desktop
5	Equipment	1	Modem
6	Estate	1	Office
7	Supplements	10	Sheets
8	Supplements	5	Folders
9	Furniture	1	Desk
10	Furniture	1	Chair
11	Hardware	1	Drone
12	Hardware	1	Cell phone
13	Hardware	2	Sensor
14	Hardware	2	Microcontroller
15	Digital	1	Receiving application
16	Digital	1	Drone management application
17	Digital	1	Sql Server
18	Digital	1	Visual Studio C#

Next, we established the tasks for the roles previously defined (see Table 5-21 to Table 5-24; Figure 5-12).

Table 5-21: Project manager tasks (The Authors).

Project manager (PM) **Responsible:** David Pulgarín	
Process name	**Task description**
PM	Update the project repository.
PM	Analyze changes in the project.
PM	Approve the project plan.
PM	Correction of deviation problems.
PM	Define the delivery instructions.
PM	Describe the elements in the project plan.
PM	Establish the work team.
PM	Establish the project repository.
PM	Estimating effort and cost.
PM	Estimate the start and end of tasks.
PM	Estimate the execution of tasks.
PM	Version control strategies.
PM	Assess project performance.
PM	Formalize the conclusion of the project.
PM	Make backup copies.
PM	Identify changes in requirements.
PM	Identify tasks.
PM	Identify and document the risks.
PM	Identify and document resources.
PM	Monitor the project plan.
PM	Review meetings with the technical leader.
PM	Review of the statement of work.
PM	Review of the project plan.

Table 5-22: Technical leader tasks (The Authors).

Technical leader (TL) **Responsible:** Jaime Alberto Guzmán Luna	
Process name	**Task description**
PM	Analyze changes in the project.
PM	Approve the project plan.
PM	Correction of deviation problems.
PM	Evaluate project performance.
PM	Identify changes in requirements.
PM	Monitor the project plan.
PM	Review meetings.
SI	Approval of the requirements specification.
SI	Assignment of tasks
SI	Review of the project plan

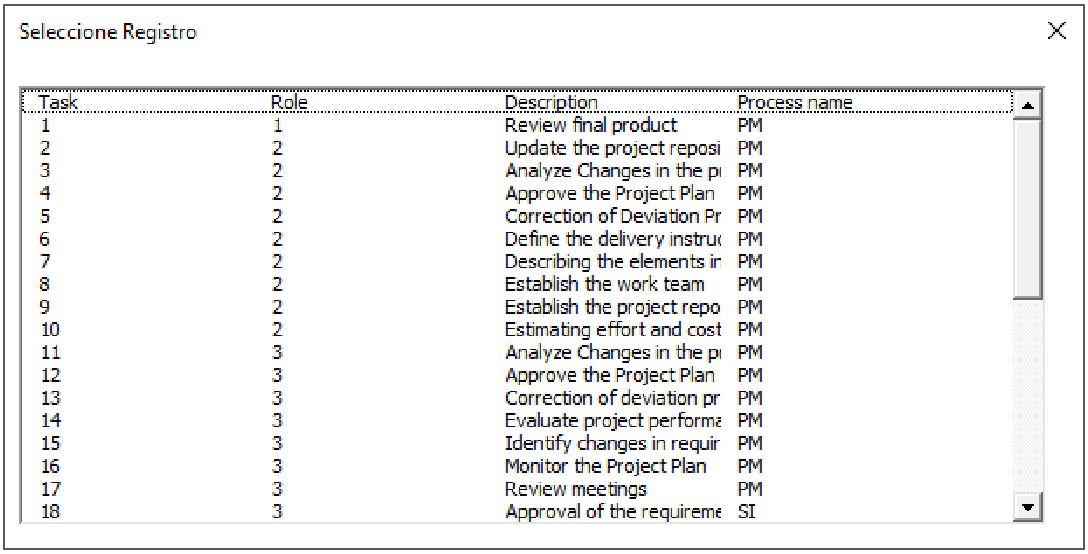
Table 5-23: Analyst tasks (The Authors).

Analyst (AN) **Responsible:** David Pulgarín	
Process name	**Task description**
SI	Elicit, specify and analyze requirements information.
SI	Designing user interfaces with ergonomic criteria.
SI	Check and edit techniques.
SI	Develop and maintain software.

Table 5-24: Client tasks (The Authors).

Client (CUS) **Responsible:** COLCIENCIAS	
Process name	**Task description**
PM	Specify product description, general requirements, and objectives.
PM	Accepts statement of work previously approved by the project manager.

Figure 5-12: Update screen of role tasks in the executable pre-conceptual schema (The Authors).


Seleccione Registro

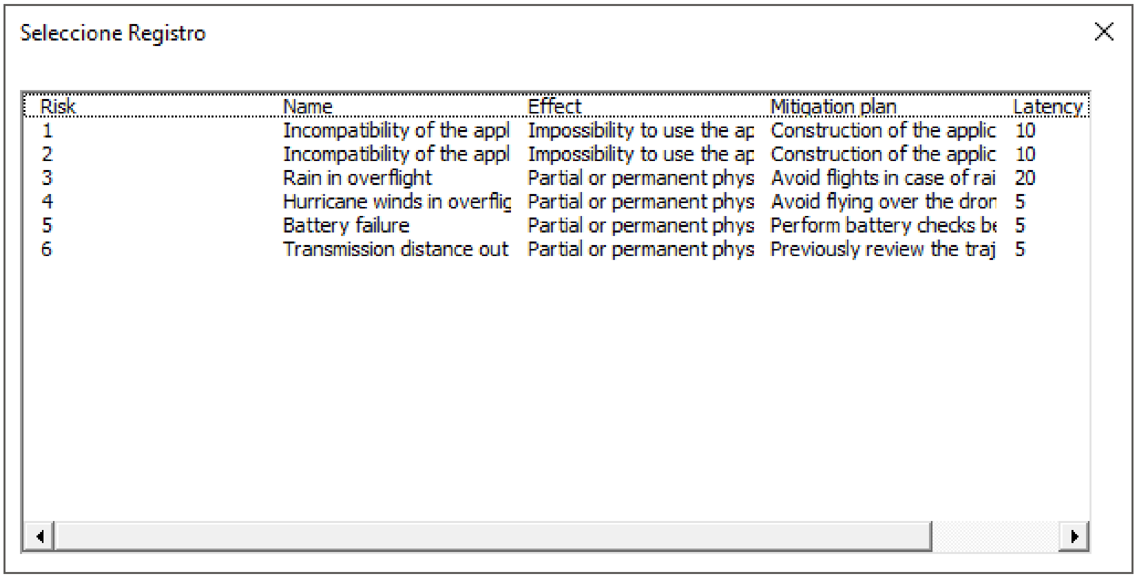
Task	Role	Description	Process name
1	1	Review final product	PM
2	2	Update the project reposi	PM
3	2	Analyze Changes in the pı	PM
4	2	Approve the Project Plan	PM
5	2	Correction of Deviation Pr	PM
6	2	Define the delivery instru	PM
7	2	Describing the elements ir	PM
8	2	Establish the work team	PM
9	2	Establish the project repo	PM
10	2	Estimating effort and cost	PM
11	3	Analyze Changes in the pı	PM
12	3	Approve the Project Plan	PM
13	3	Correction of deviation pr	PM
14	3	Evaluate project performa	PM
15	3	Identify changes in requir	PM
16	3	Monitor the Project Plan	PM
17	3	Review meetings	PM
18	3	Approval of the requireme	SI

Finally, we document the possible project risks specifying the name, effect, mitigation actions, and probability (see Table 5-25; Figure 5-13).

Table 5-25: Project risks (The Authors).

Risk	**Effect**	**Mitigation actions**	**Probability**
Incompatibility of the application with Android versions.	Impossibility to use the application correctly.	Construction of the application in the largest existing universal version.	10%
Incompatibility of the application with hardware.	Impossibility to use the application correctly	Construction of the application with a reduced use of resources.	10%
Rain in overflight.	Partial/permanent physical damage to the drone.	Avoid flights in case of rain forecast.	20%
Hurricane winds in overflight.	Partial/permanent physical damage to the drone.	Avoid flying during hurricane season (June, July, and August, from 2:00 p.m. to 6:00 p.m.).	5%
Battery failure.	Partial/permanent physical damage to the drone.	Perform battery checks before each flight.	5%
Transmission distance out of range.	Partial/permanent physical damage to the drone.	Previously review the trajectories so the data is not lost due to transmission distance faults.	5%

Figure 5-13: Update screen of the project risks identified in the executable pre-conceptual schema (The Authors).



Risk	Name	Effect	Mitigation plan	Latency
1	Incompatibility of the appl	Impossibility to use the ap	Construction of the applic	10
2	Incompatibility of the appl	Impossibility to use the ap	Construction of the applic	10
3	Rain in overflight	Partial or permanent phys	Avoid flights in case of rai	20
4	Hurricane winds in overflic	Partial or permanent phys	Avoid flying over the dror	5
5	Battery failure	Partial or permanent phys	Perform battery checks be	5
6	Transmission distance out	Partial or permanent phys	Previously review the traj	5

By completing the work products, we had the tools for evaluating the project viability. Besides, with such criteria, we can decide whether we can proceed to a further meeting with the customer in order to negotiate cost, delivery time, and the other work products included in the project plan. The next meeting with the customer included the agreement of the statement of work developed by establishing an accurate project domain for continuing the project management process.

6 Conclusions and future work

6.1 Conclusions

In this M.Sc. Thesis, we proposed a method in the context of the ISO/IEC 29110 basic profile for eliciting requirements in very small entities based on pre-conceptual schemas. We characterized five main concepts in the project management and software implementation processes regarding requirements elicitation. Furthermore, we developed an executable pre-conceptual schema based on the work products previously established for implementing the proposed method.

In the case study, we showed the execution of the complete requirements elicitation cycle in a real-life research project. Fluency and naturalness in the meetings with the customer denoted the benefit generated by the usage of a tool with cognitive and interactive features. The graphical environment provided by the executable pre-conceptual schema acted as a map for guiding the user in the requirements elicitation process. Besides, such an environment allowed the user for navigating inside the project domain, fulfilling the tasks specified in each work product defined in the standard. The resulting product is an unambiguous, lightweight, and accurate method with high proximity to the stakeholder language. However, the method demands the requirements specification work product to be performed partially in early stages of the project. Such a work product includes the development of knowledge representations of the domain and remaining product characteristics (reliability, efficiency, maintenance, portability, design limitations, interoperability, reusability, and legal aspects) at the discretion of the project manager. With overloads of knowledge representations and highly populated characteristic specifications, the method could lack effectiveness in terms of cost and time.

The ISO/IEC 29110 includes a set of profiles, processes, activities, and tasks accepted as best practices for improving quality in software and hardware projects, establishing the what-to-do when developing such projects. The proposed method in this M.Sc. Thesis emphasized the accurate requirements elicitation by implementing executable pre-conceptual schemas based on such accepted practices, providing the how-to-do-it when implementing such a standard. We attempt to develop an alternative regarding VSEs when implementing the ISO/IEC 29110 with this approach.

6.2 Future work

Further study should consider the implementation of the method based on pre-conceptual schemas used in this M.Sc. Thesis for representing other components included in the basic profile of the ISO/IEC 29110. The representation of the project planning, the project plan execution, the project assessment and control, and the project closure activities described in the project management process. We proposed as well the representation of the remaining activities and tasks included in the software implementation process. Therefore, the basic profile could be entirely represented in order to provide a viable tool for VSEs when implementing and certificating ISO/IEC 29110.

As a second line of future work, we propose the representation of ISO standards related to the ISO/IEC 29110 by using the method proposed in this M.Sc. Thesis. Standards such as ISO/IEC 12207 (Information Technology/Software Life Cycle Processes) and ISO/IEC 29119 (Software and systems engineering/Software testing) should be represented for harmonization purposes. A main domain containing common and specific objects of such standards should be constructed in order to obtain a macrostructure based in a knowledge representation for jointly executing such standards.

Finally, we propose for future work an increasing number of case studies by using the method proposed in this M.Sc. Thesis. VSEs with different environments, lifecycles, countries, etc. should evaluate the approach proposed. The method proposed could be modified in pursuit of meeting real VSEs needs by collecting information, experiences, facts, and results.

Bibliography

Abarca, M. Arisaca, R. & Dávila A. (2015). Implementación del Perfil Básico de la ISO/IEC 29110 de una Pequeña Empresa Desarrolladora de Software: Lecciones Aprendidas. Industrial (INDUSTRIAL 2015) en XVIII Congreso Ibero Americano de Ingeniería de Software (CIBSE 2015). Lima.

Bougaa, M. O'Connor, R. V. Bornhofen, S. & Rivière, A. (2017). A Standard Based Adaptive Path to Teach Systems Engineering: 15288 and 29110 Standards Use Cases. 11th Annual IEEE International Systems Conference, Montréal.

Buchalcevova, A. (2017). Methodology for ISO/IEC 29110 Profle Implementation in EPF Composer. International Journal of Information Technologies and Systems Approach, vol. 10, no. 1, pp. 61-74.

Díaz, A. de Jesús, C. Melendez, K. & Dávila, A. (2016). ISO/IEC 29110 Implementation on two Very Small Software Development Companies in Lima. Lessons Learned. IEEE Latin America Transactions, vol. 14, no. 5, pp. 2504-2510.

Chaverra, J. (2011). Generación automática de prototipos funcionales a partir de esquemas preconceptuales (M.Sc. thesis). Universidad nacional de Colombia, Colombia.

Eito, R. (2014). Ontology-based Tailoring of Software Process Models. Terminology and Knowledge Engineering. Berlin.

Galvan, S. Mora, M. O'Connor, R. V. Acosta, F. & Alvarez, F. (2015). A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. Procedia Computer Science, vol. 64, pp. 188-195.

García, L. Laporte, C. Y. Arteaga, Y. & Bruggmann, M. (2015). Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. Software Quality Professional, vol. 17, no. 2, pp. 16-29.

Hidalgo, D. (2009). Recopilación y ejemplificación de la especificación completa de los esquemas preconceptuales. (Bachelor thesis). Universidad Nacional de Colombia, Colombia.

ISO/IEC. (2016). ISO/IEC TR 29110-1: 2016 - Software Engineering - Lifecycle profiles for Very Small Entities (VSES) - Part 1: Overview. Geneva.

ISO/IEC. (2011). ISO/IEC TR 29110-5-1-2: 2011 - Software Engineering - Lifecycle profiles for Very Small Entities (VSES) - Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. Geneva.

Laporte, C. Y. (2011). The Development and Experimentation of an International Standard for Very Small Entities Involved in Software Development. International Council on Systems Engineering (INCOSE) Workshop, Phoenix, AZ.

Laporte, C. Alexandre, S. & O'Connor, R. V. (2008). A Software Engineering Lifecycle Standard for Very Small Enterprises. European Conference on Software Process Improvement EuroSPI 2008, Dublin, vol. 16, pp. 129–141.

Laporte, C. Y. & O'Connor, R. V. (2016). Implementing Process Improvement in Very Small Enterprises with ISO/IEC 29110 A Multiple Case Study Analysis. 2016 10th International Conference on the Quality of Information and Communications Technology, Lisbon, pp. 125-130.

Laporte, C. Y. Séguin, N. Boas, G. V. & Buasung, S. (2013). Small tech firms Seizing the benefits of software and systems engineering standards. Rev. ISO Focus+, vol. 4, no. 2, pp. 32-36.

Laporte, C. Y. Tremblay, N. Menaceur, J. & Poliquin, D. (2016). Implementing the New ISO/IEC 29110 Systems Engineering Process Standard in a Small Public Transportation Company. Systems, Software and Services Process Improvement EuroSPI 2016, C. Kreiner, R. V. O'Connor, A. Poth and R. Messnarz, eds., Springer-Cham, Graz, pp. 15-29.

Manrique, B. & Zapata, C.M. (2014). Transformación de lenguaje natural a controlado en la educción de requisitos: una síntesis conceptual basada en esquemas preconceptuales, Rev. Fac. Ing. Univ. Antioquia, no. 70, pp. 132-145.

O'Connor, R. V. (2014). Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices: A Case Study. Software Process Improvement and Capability Determination, 14th International Conference, SPICE 2014 Vilnius. Communications in Computer and Information Science. Lithuania, A. Mitasiunas, T. Rout, R. O'Connor and A. Dorling, eds., Springer-Cham, vol. 477, pp. 226-237.

Organisation for Economic Co-operation and Development. (2005). OECD SME and Entrepreneurship Outlook - 2005 Edition. OECD Publishing, Paris.

Rodríguez-Dapena, P. & Buitrago-Botero, M. F. (2015). How to Certify the Very Small Entity Software Processes Using ISO/IEC 29110. Communications in Computer and Information Science 526 - Software Process Improvement and Capability Determination - 15th International Conference, SPICE 2015. Gothenburg, pp 245-252.

Sánchez-Gordón, M. L. & O'Connor, R. V. (2016). Understanding the gap between software process practices and actual practice in very small companies. Software Quality Journal, vol. 24, no. 3, pp. 549-570.

Sanchez-Gordon, M. L. O'Connor, R. V. & Colomo-Palacios, R. (2015). Evaluating VSEs Viewpoint and Sentiment towards the ISO/IEC 29110 Standard: A two Country Grounded Theory Study. Software Process Improvement and Capability Determination SPICE 2015, T. Rout, R. V. O'Connor and A. Dorling, eds., Springer-Verlag, Gothenburg, pp. 114-127.

Siddoo, V. Wongsai, N. & Wetprasit, R. (2013). An Implementation Approach of ISO/IEC 29110 for Government Organizations. Lecture Notes in Computer Science, vol. 7983, pp. 5-19.

Zapata, C.M. & Arango, F. (2005). Los modelos verbales en lenguaje natural y su utilización en la elaboración de esquemas conceptuales para el desarrollo de software: una revisión crítica. Revista EIA, vol. 41, no. 137, pp. 77-95.

Zapata, C. M. (2007a). Definición de un esquema pre-conceptual para la obtención automática de esquemas conceptuales de UML (P.h.D thesis). Universidad Nacional de Colombia, Colombia.

Zapata, C. M. Gelbukh, A. & Arango, F. (2006a). Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation. Research in Computing Science: Advances in Computer Science and Engineering, vol. 4293. pp. 17-27.

Zapata, C. M. Gelbukh, A. & Arango, F. (2006b). Pre-conceptual Schema: a UML Isomorphism for Automatically Obtaining UML Conceptual Schemas. Research in Computing Science: Advances in Computer Science and Engineering, vol. 19, pp. 3–13

Zapata, C. M. Giraldo, G. & Londoño, S. (2011). Esquemas preconceptuales ejecutables. Revista Avances en Sistemas e Informática, vol. 17, no. 2, pp. 15– 24

Zapata, C. M. Lezcano, L.A. & Tamayo, P.A. (2007b). Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales. Revista EIA, no. 8, pp. 21-35.

Zapata, C. M. Vargas, F. (2013). Reglas Sintáctico-semánticas para Relacionar los Objetivos Organizacionales y los Problemas en el Contexto de la Educción Temprana de Requisitos de Software. Revista latinoamericana de ingeniería de software, vol. 1, pp. 01-07.

Zapata, C.M. Villa, H. & Chaverra J. (2014). Reglas heurísticas para la generación de casos de uso a partir de esquemas preconceptuales. Cuaderno Activa, vol.6, pp19-31.

Wongsai, N. Siddoo, V. & Wetprasit, R. (2015). Factors of Influence in Software Process Improvement: An ISO/IEC 29110 for Very-Small Entities. 2015 7th International Conference on Information Technology and Electrical Engineering, Chiang Mai, pp. 12-17.