

Requirements engineering base process for a quality model in Cuba

Yoandy Lazo Alvarado  | Centro Nacional de Calidad de Software (CALISOFT) | yoandy.lazo@calisoft.cu |
Leanet Tamayo Oro  | Centro Nacional de Calidad de Software (CALISOFT) | leanet.tamayo@calisoft.cu |
Odannis Enamorado Pérez  | Centro Nacional de Calidad de Software (CALISOFT) odannis.enamorado@calisoft.cu |
|
Karine Ramos  | ALLOY - Digital Product Development & Marketing Technology | karine.rb19@gmail.com |

Abstract

A high percentage of projects worldwide fail or are canceled, due to incorrect requirements engineering. Incorporating good practices into the requirements engineering process provides the appropriate mechanism to understand and analyze what stakeholders want and need. This process also allows you to evaluate and negotiate a reasonable solution; and specify, validate and manage the requirements as they become a functional system. The objective of this research is to elaborate a process of Requirements Engineering for the Quality Model for Software Development that contributes to raising the percentage of successful projects in Cuban's software development organizations, regarding the fulfillment of the agreed requirements. To reach the desired goal a bibliographic review was made about the requirements engineering discipline, as well as interviews and surveys to roles related to this activity in Cuban's software development organizations. The solution was evaluated by experts, in a focus group and put into practice, as a pilot, in three organizations. As a result, a basic Requirements Engineering process was obtained that contains specific requirements divided by the three levels of maturity of the model, and a graphic and textual description of the process. The satisfaction of the end user was measured through the implementation of Iadov technique, obtaining a Group Satisfaction Index equals 1, meaning maximum user's satisfaction with the process.

Keywords: requirement, requirements engineering, software, process

1. Introduction

A significant percentage of software development projects, worldwide, are canceled or fail according to studies carried out. Between the years 2011 and 2015 those canceled accounted for 39%, 46%, 40%, 47% and 45%, respectively, and the unsuccessful represented 22%, 17%, 19%, 17% and 19%, respectively (Rosato, 2018; The Standish Group International, 2015). The behavior of projects in 2018 was similar to previous years since canceled projects reached 36% and 20% reported as unsuccessful (International, 2018). An investigation carried out by Lehtinen et al. suggests that the causes of failures in projects occur in several processes that include management, sales and requirements, and implementation. It also states that the failures are related to the project environment, people, methods, and tasks (Lehtinen, Mäntylä, Vanhanen, Itkonen, & Lassenius, 2014; McLeod & MacDonell, 2011).

An analysis of the Standish Group publication in 2014 on a study of more than 2,000 projects in 1,000 companies allowed to know that, although project is considered successful regarding the compliance with delivery deadlines, budget and agreed requirements, the percentages of utilization of functionalities that compose the systems are: 7% always, 13% often, 16% sometimes, 19% rarely, 45% never (The Standish Group International, 2014). According to del Toro, this is because: 1) The client did not request it, but it could happen due to a misinterpretation of a requirement or the developers considered that it could be useful or interesting; or 2) The client requested it, i) but

later he realized that he did not describe it correctly and he does not want it anymore; ii) the client described it correctly, but when he saw it implemented he realized that he asked for something wrong; or iii) the client described it correctly, but now wants something different (del Toro, 2018).

The authors of this research agree with del Toro on the importance of maintaining adequate feedback with stakeholders during the software development life cycle to reduce the effects of the volatility of the requirements. To guarantee that feedback, an indispensable bridge that goes through the stakeholder needs, the design and development of the product is the Requirements Engineering (RE) process. It provides an appropriate mechanism to understand and analyze the stakeholders needs, evaluate the feasibility, negotiate a reasonable solution, specify the solution without ambiguities, validate the specification and manage the requirements as they become a functional system (Pressman, 2010).

A diagnosis performed in 2014 by the Software Quality National Center (CALISOFT) to a sample of 43.75% of Cuban's software development organizations, allowed to characterize these organizations through the application of interviews and surveys to the roles involved in the RE process (CALISOFT, 2014). Among the evaluated aspects, one of them is the fulfillment of the activities that compose the RE process proposed by Pressman (Pressman, 2010). An analysis of the obtained results allowed the authors of this research to know that 7.14% do not identify the stakeholder requirements, 14.26% do not specify them, 21.43% do not validate them, 28.57% do not implement

changes control, and 57.14% do not maintain traceability with the requirements. Another result obtained describes the behavior of the projects completed in the period 2011-2014, identifying that 16.42% of the projects did not complete all the requirements agreed with the client; 11.94% delivered out of time and did not complete all the agreed requirements; 1.49% did not complete all the requirements and were over budget (CALISOFT, 2014; Pérez & Aveleira, 2016).

Another diagnosis performed in 2017 to a sample of 28.13% of Cuban's software development organizations. It allowed to know that 48% of the completed projects did it successfully, 20% canceled, and 32% failed. It also allowed identifying that the percentage of implementation of the RE process reached 48% (CALISOFT, 2017).

For all the above described it can be stated that the process in question is not mature and recognizes the need to establish activities that provide more feedback with the client.

The software development organizations in Cuba have used the NC-ISO 9001, and CMMI to reach maturity levels in their development processes (Y. A. Lazo, 2016). At the same time, CALISOFT researchers work on the development of the Quality Model for the Development of Computer Applications (MCDAI) to provide the industry with a model based on international best practices.

MCDAI takes into account national characteristics and is based on the following principles: easy to understand, easy to apply, and serve as a basis for evaluations in other internationally recognized models (Pérez, 2014).

That being said the objective of this research is to develop a process of Requirements Engineering for the MCDAI that contributes to raise the percentage of successful projects in Cuban's software development organizations, regarding compliance with the agreed requirements.

2. Theoretical framework

2.1. Requirements engineering in software development

As part of the construction of the theoretical framework of the research, a bibliographic review was carried out (Goguen, 1994; IEEE, 2014; ISO, IEC, & IEEE, 2017; Oficina Nacional de Normalización, 2015b; Sommerville, 2011; Team, 2010), this allowed to conceptualize the term *requirements* as: need or expectation established, generally implicit or mandatory, expressing a condition or capacity demanded by the stakeholders or the organization, which must comply or have a process, product or product component to solve a problem or achieve an objective and to satisfy a contract, standard, specification or other formally imposed document.

The broad spectrum of tasks and techniques that lead to understanding requirements is called RE. From the software process perspective, RE is one of the software engineering important actions, that begins during the communication activity and continues in the modeling

activity. Pressman argues that as part of the RE seven different tasks are performed which are: conception, inquiry, elaboration, negotiation, specification, validation, and management (Pressman, 2010). However, Sommerville identifies that the main activities of RE are the acquisition, analysis, and validation of requirements. He also explains the importance of the requirements administration to plan the RE process activities and control requirements changes (Sommerville, 2011).

The Guide to the Software Engineering Body of Knowledge (SWEBOK) contains the Software Requirements knowledge area (KA) that is concerned with the elicitation, analysis, specification, and validation of software requirements as well as the management of requirements during the whole life cycle of the software product (IEEE, 2014).

2.1.1. Requirements engineering according to NC-ISO 9001

NC-ISO 9001:2015 Quality management systems - Requirements, uses the process approach, which incorporates the Plan-Do-Verify-Act cycle and risk-based thinking. The organizations that use it do so with a strategic vision to improve their overall performance. This standard can be used in any organization, including those that develop software.

In this standard, the RE process is not explicitly delimited, but it states that "the organization must plan, implement and control the processes necessary to meet the requirements for the provision of products and services, and implement the determined actions"; the aforementioned gives the organization a possibility to implement a RE process for software development. Also, it raises several requirements related to the RE process: "8.2.2 Determining the requirements for products and services", "8.2.3 Review of the requirements for products and services", and "8.2.4 Changes to requirements for products and services" (Oficina Nacional de Normalización, 2015a).

Organizations that develop software can use ISO/IEC/IEEE 90003:2018 as a guide for the application of ISO 9001. ISO/IEC/IEEE 90003 explains in detail how to comply with the requirements mentioned above (ISO, IEC, & IEEE, 2018).

2.1.2. RE according to ISO/IEC/IEE 12207 and ISO/IEC/IEEE 15288

ISO/IEC/IEEE 15288:2015 and ISO/IEC/IEEE 12207:2017 contain the life cycle processes for the system and software, respectively. Both international standards contain 30 processes, including two related to RE. During the revision of these standards, it was possible to identify that the activities they propose for ER are similar.

1. The purpose of the **Stakeholder Needs and Requirements Definition process** is to define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment (ISO, IEC, & IEEE, 2015; ISO et al., 2017).

The project to declare full compliance with this process shall implement the following activities:

- a) Prepare for Stakeholder Needs and Requirements Definition;
 - b) Define stakeholder needs;
 - c) Develop the operational concept and other life cycle concepts;
 - d) Transform stakeholder needs into stakeholder requirements;
 - e) Analyze stakeholder requirements; and
 - f) Manage the stakeholder needs and requirements definition.
2. The purpose of the **System/Software Requirements Definition** process is to transform the stakeholder, user-oriented view of desired capabilities into a technical view of a solution that meets the operational needs of the user. (ISO et al., 2015, 2017).
- The project to declare full compliance with this process shall implement the following activities:
- a) Prepare for System/Software Requirements Definition;
 - b) Define system/software requirements;
 - c) Analyze system/software requirements; and
 - d) Manage system/software requirements.

2.1.3. RE according to Capability Maturity Model Integration (CMMI)

CMMI is a process improvement maturity model for the development of products and services. It includes the best practices that deal with development and maintenance activities that cover the product's life cycle, from conception to delivery and maintenance. It was created by the Software Engineering Institute at Carnegie Mellon University and is the result of the integration of several models (CMMI Institute, 2015).

CMMI has 22 process areas. A process area is a set of related practices that when implemented collectively, satisfy a set of objectives considered important to improve that process area (CMMI Institute, 2015). The process areas are composed of specific goals (SG) and specific practices (SP) that guide in a more detailed way how to achieve the goals. Two of the model areas work on the topic of software requirements: Requirements Development (RD) and Requirements Management (REQM).

The purpose of the **RD process area** is to elicit, analyze, and establish customer, product, and product component requirements (Team, 2010). This process area includes three SG.

SG 1 Develop customer requirements.

SP 1.1 Elicit needs.

SP 1.2 Transform stakeholder needs into customer requirements.

SG 2 Develop product requirements.

SP 2.1 Establish product and product component requirements.

SP 2.2 Allocate product component requirements.

SP 2.3 Identify interface requirements.

SG 3 Analyze and validate requirements.

SP 3.1 Establish operational concepts and scenarios.

SP 3.2 Establish a definition of required functionality and quality attributes.

SP 3.3 Analyze requirements.

SP 3.4 Analyze requirements to achieve balance.

SP 3.5 Validate requirements.

The purpose of the **REQM process area** is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products (Team, 2010). This process area is composed of a single SG.

SG 1 Manage requirements.

SP 1.1 Understand requirements.

SP 1.2 Obtain commitment to requirements.

SP 1.3 Manage requirements changes.

SP 1.4 Maintain bidirectional traceability of requirements.

SP 1.5 Ensure alignment between project work and requirements.

2.1.4. RE according to Brazilian software process improvement (MPS.Br)

MPS.Br was created by the Association for the Promotion of the Excellence of Brazilian Software (SOFTEX). It has three components: Reference Model (MR-MPS), Evaluation Method (MA-MPS) and Business Model (MN-MPS). It is composed of 19 process areas. It focuses on the profile of companies with different sizes and characteristics, although with special attention to micro, small, and medium enterprises. The model has two areas that address RE: Requirements Development (DRE) and Requirements Management (GRE) (Montoni, Rocha, & Weber, 2009).

The purpose of the **DRE process area** is to define customer, product, and product components requirements. The expected results of DRE process are (SOFTEX, 2009a):

- DRE1 - The client needs, expectations, and restrictions, both of the product and its interfaces, are identified.
- DRE2 - A defined set of customer requirements is specified based on needs, expectations, and restrictions identified.
- DRE3 - A set of product and product components functional and non-functional requirements that describe the problem solution to be solved is defined and maintained based on the client requirements.
- DRE4 - Each product component functional and non-functional requirements are refined, elaborated, and designated.
- DRE5 - Product and each product component internal and external interfaces are defined.
- DRE6 - Operating concepts and scenarios are developed.

- DRE7 - The requirements are analyzed using defined criteria to balance stakeholder needs with the existing restrictions.
- DRE8 - The requirements are validated.

The purpose of the **GRE process area** is to manage product and product components requirements of the project, and identify inconsistencies between requirements and project plans and project work products. The expected results of GER process are (SOFTEX, 2009b):

- GRE1 - The requirements are understood, evaluated, and accepted together with the requirements providers, using objective criteria.
- GRE2 - The technical team's commitment to the approved requirements is obtained.
- GRE3 - Bidirectional traceability between requirements and work products is established and maintained.
- GRE4 - Revisions in plans and work products of the project are carried out to identify and correct inconsistencies with the requirements.
- GRE5 - Requirements changes during the project are managed.

2.1.5. RE according to MoProsoft and COMPETISOFT

Software Industry Processes Model (MoProSoft) emerges as part of the Ministry of Economy Software Industry Development Program from Mexico, to reach international levels of process capacity by small and medium-sized Mexican software development companies (Hanna Oktaba, 2015). This model was the basis for the preparation of the ISO/IEC 29110 – Lifecycle profiles for Very Small Entities, and the model Process Improvement to Promote the Competitiveness of the Ibero-American Small and Medium Software Industry (COMPETISOFT) (COMPETISOFT, 2006).

MoProSoft and COMPETISOFT are divided into three categories Senior Management, Management, and Operation. The Operations category contains the Software Development and Maintenance process, which allows to systematically carrying out requirements engineering activities, with a set of activities whose purpose is to obtain the documentation of the requirement specification and the system, to have a common understanding between the customer and the project. Some of these activities for

MoProSoft model are (COMPETISOFT, 2006; Hanna Oktaba, 2005):

A2.2. Document or modify Requirement Specifications.

- Identify and query information sources (customers, users, previous systems, documents, etc.) to obtain new requirements.
- Analyze requirements identified to limit their scope and feasibility, considering the customer's or project's business environment restrictions.
- Prepare or modify the user interface prototype.
- Generate or update the Requirement Specifications.

A2.3. Verify the Requirements Specification.

A2.4. Correct defects found in Requirement Specification based on Verification Report and obtain approval of corrections.

A2.5. Validate Requirements Specification.

A2.6. Correct defects found in Requirements Specification based on Validation Report and obtain approval of corrections.

A3.2. Document or modify Analysis and Design.

- Generate or modify the Traceability Record.

COMPETISOFT incorporates other activities that complex MoProSoft, e.g., A2.2 includes the task to identify and establish the security requirements of the information standard to obtain the required level of security. In general way, they both describe similar activities for requirements engineering.

2.1.6. Good practices extracted from the models and standards

RE has a fundamental role in software development projects because it is the process that allows communication with stakeholders to obtain the requirements of the product in development. Some models and standards group RE activities in requirements development and requirements management. After analyzing the bibliography studied, it can be affirmed that CMMI and MPS.Br treats similarly the RE. The same way happens with MoProSoft and COMPETISOFT, as well as ISO/IEC/IEEE 12207 and 15288 standards.

Table 1 identifies the good practices of the RE. The requirements elicitation, specification, analysis, and validation stand out in the requirements development. In the case of requirements management, the most common is to achieve understanding, control changes, and maintain bidirectional traceability.

Table 1. Good practices in the RE process (own preparation).

No.	Good practices	Pressman	Sommerville	NC-ISO 9001	ISO/IEC/IEEE 12207 y 15288	CMMI – DEV y MPS. Br	MoProSoft y COMPETISOFT	SWEBOK
Requirements Development								
1	Requirements elicitation	5.1 –	c4s5	4.2	6.4.2.3	RD	9.2(OPE.2-	Chapter

No.	Good practices	Pressman	Sommerville	NC-ISO 9001	ISO/IEC/IEEE 12207 y 15288	CMMI – DEV y MPS. Br	MoProSoft y COMPETISOFT	SWEBOK
		inquiry 5.3		5.1.2	(b [1, 2, 3, 4]) 6.4.2.3 (d [1, 2, 3])	(SG 1 [SP 1.1, SP 1.2]) DRE 1	A2.2) DS(A2.2)	(3)
2	Requirements specification	5.1 – specification	c4s2		6.4.3.3 (b [3, 4, 5])	RD (SG 2 [SP 2.1, SP 2.2]) DRE 2 DRE 4	9.2(OPE.2- A2.2, A3.2) DS(A2.2, A3.2)	Chapter (5)
3	Elaboration of product requirements		c4s5		6.4.3.3 (b [1, 2, 3])	RD (SG 2 [SP 2.1, SP 2.2]) DRE 3	DS(A3.2)	Chapter (4.2)
4	Identify interface requirements	5.1 – elaboration 5.4 7			6.4.3.3 (b [5])	RD (SG 2 [SP 2.3]) DRE 5	9.2(OPE.2- A2.2, A3.2) DS(A2.2, A3.2)	
5	Establish operational concepts and associated scenarios	5.1 – elaboration 5.5 6	c4s5		6.4.2.3 (c [1, 2])	RD (SG 3 [SP 3.1]) DRE 6	DS(A3.2)	Chapter (4.2)
6	Establish a definition of required functionality and quality attributes		c4s1		6.4.3.3 (b [5])	RD (SG 3 [SP 3.2])	DS(A2.2)	Chapter (7.3)
7	Analysis and negotiation	5.1 negotiation 5.6	c4s5 c12s5	8.2.1 8.2.3	6.4.2.3 (e [1, 2, 3, 4]) 6.4.3.3 (c [1, 2, 3, 4])	RD (SG 3 [SP 3.3, SP 3.4]) DRE 7	9.2(OPE.2- A3.2) DS(A2.2, A3.2)	Chapter (4)
8	Validation of requirements	5.1 – validation 5.7	c4s6			RD (SG 3 [SP 3.5]) DRE 8	9.2(OPE.2- A2.5, A2.6) DS(A2.5, A2.6)	Chapter (6)
Requirements management								
9	Identify requirements source	5.1 - management	c4s5	8.1	6.4.2.3 (a [1, 2,]) 6.4.3.3 (a [1, 2,])	REQM (SG 1 [SP 1.1])	9.2(OPE.2- A2.2) DS(A2.2)	Chapter (3.1)

No.	Good practices	Pressman	Sommerville	NC-ISO 9001	ISO/IEC/IEEE 12207 y 15288	CMMI – DEV y MPS. Br	MoProSoft y COMPETISOFT	SWEBOK
10	Understand requirements			8.2.3		REQM (SG 1 [SP 1.1]) GRE 1		
11	Obtain commitment to requirements			8.2.1	6.4.2.3 (f[1]) 6.4.3.3 (d [1])	REQM (SG 1 [SP 1.2]) GRE 2		
12	Manage requirements changes		c4s7	8.2.4	6.4.2.3 (f[3]) 6.4.3.3 (d [3])	REQM (SG 1 [SP 1.3]) GRE 5		Chapter (7.2)
13	Maintain traceability of requirements				6.4.2.3 (f[2]) 6.4.3.3 (d [2])	REQM (SG 1 [SP 1.4]) GRE 3	DS(A2.2, A3.2, A4.3, A4.6)	Chapter (7.4)
14	Ensure alignment between project work and requirements			8.2.4		REQM (SG 1 [SP 1.5]) GRE 4	9.2(OPE.2- A1.1) DS(A1.1, A2.2, A2.3)	

The tendency to include the reuse management approach as a process of creating software systems from existing software, applying domain engineering, was also identified in these reference models. This approach has provided many organizations with competitive advantages in the market, in terms of product quality, development time, production costs, among others (Bastarrica, 2011; Manso Martínez & García Peñalvo, 2013; Northrop et al., 2007; Salazar, 2017).

During the application of domain engineering requirements are also developed and managed. As a fundamental element of domain engineering is the application of the domain analysis technique. It allows capturing the critical information of the entities, data, and processes that characterize a particular business area and then develop and specify the requirements (Brun, 2007). The main result of the application of this technique is the domain model, which describes at a high level of abstraction the common elements and variants of the family for a correct management of the variability of the resulting products (Montoni et al., 2009).

Most of the studied models and standards are designed for large software development organizations since they need long periods of implementation, and great effort for their assimilation. Also, they have a high cost associated

with certification and consulting, so it is difficult for Cuban organizations, which have limited resources, to adopt some of these.

It is for this reason that countries that are characterized by the majority presence of Small and Medium Enterprises (SME) such as Mexico and Brazil have adapted the internationally recognized models to their needs, like MOPROSOFT in the case of Mexico and MPS for the Brazilian development companies. However, these two projects are adapted to the context of these countries and their characteristics. Most of the available models do not detail a strategy that allows organizations an agile process that guides improvement and facilitates the work of process engineers.

2.2. Model for the Development of Computer Applications

The processes' capacity to adapt to the market or clients makes that management models, oriented to quality, focus their attention on processes as the most powerful lever to act on the results, in an effective and sustained way in a long time (Concepción, 2010; Zaratiegui, 1999). Perez researcher states that the MCDAI has a process approach and considers it an accepted proposal for the software development industry in Cuba (Pérez, 2014).

The MCDAI is composed of 1) **General Guide** that describes the model and its components; 2) **Implementation Guide** that contains the general requirements that must be met by the twelve base processes that compose the model, as well as defining each of the base processes; 3) **Evaluation Guide** that describes the process and evaluation method to determine the organization maturity level and capacity of its processes related to the model (see Figure 1) (Pérez, 2014).



Figure 1. MCDAI components.

Implementation Guide groups the processes into the following categories 1) **Organizational Management** gathers the base processes that have a direct influence on the organization, and it executes at a high level or on Management's responsibility; 2) **Project Management** gathers the base processes related to the project work organization; 3) **Engineering** gathers the technical base processes necessary for software development; 4) **Support** gathers the base processes that supports software development (see Figure 2).



Figure 2. MCDAI Categories.

Each base process contains a purpose, specific requirements, and a process modeling suggested that meets the requirements. In the case of specific requirements, they

are defined by Basic, Intermediate, and Advanced levels (see Figure 3).

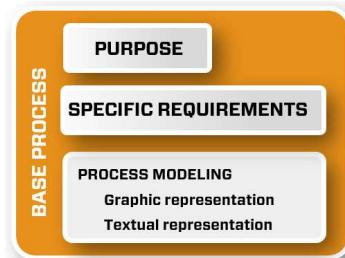


Figure 3. Base process structure.

The specific requirements are divided into three parts: **title**, **description**, and **recommended evidence**. The recommended evidences are examples of what the work products could be.

The specific requirements of each base process and MCDAI's generic requirements are used as a reference standard by evaluators to determine the organization's processes capacity. The organization's maturity level (Basic, Intermediate, or Advanced), is determined by taking into account the capacity of all its processes. Organizations that decide to adopt the MCDAI shall implement the requirements depending on the maturity level, and/or capacity desired.

The process modeling suggestion with a graphic and textual representation is also shown as part of each base process. This process modeling is done to exemplify how to implement generic and specific requirements.

2.2.1. MCDAI's generic requirements

Table 2 shown MCDAI's generic requirements (GR) necessary to reach the desired capacity. Each base process has to implement these requirements including RE Base Process, that this investigation is presenting.

Table 2. MCDAI's generic requirements.

Basic level

- GR 1 Define the process to follow.
- GR 2 Define roles and responsibilities.
- GR 3 Plan process execution.
- GR 4 Provide resources.
- GR 5 Monitor process execution.
- GR 6 Identify and preserve the configuration items.
- GR 7 Evaluate the execution of the established process.
- GR 8 Analyze the process status with the management.

Intermediate level

- GR 9 Institutionalize the process.
- GR 10 Manage indicators.
- GR 11 Train staff.
- GR 12 Manage the knowledge generated by the process.
- GR 13 Identify and treat risks.

Advanced level

- GR 14 Perform process improvement.

2.3. Process representation

To model the RE base process is necessary to analyze graphic or textual representation techniques: flow diagram, notation lanes, IDEF, ETVX, business process modeling notation (BPMN), and textual description (Losavio, Guzmán, & Matteo, 2011; MANENE, 2013; Medina, 2012; Murcia-Oeste-Arrixaca, 2013; Silega, 2014; Batista Anisbert Suárez, 2013).

The result of this analysis allowed the authors of this investigation to determine that the BPMN and textual description combination are the most optimal variant because: they allow a graphic notation that describes the logic of the steps of a business process; coordinates the sequence of processes and messages that flow between the participants of the different activities; allows processes modeling in a unified and standardized way which facilitates an understanding to everybody in the organization; explains the activities and covers the information about the needs of the process, when it begins, the people involved, the duration, how the activities are carried out, when it ends, and the different scenarios that may arise (Y. A. Lazo, 2016).

3. Requirements Engineering Base Process

This research proposes the RE base process. It is part of the MCDAI, therefore it's aligned to its structure.

3.1. Purpose and specific requirements

The purpose of the RE base process is to identify the stakeholder requirements for a software product so that it can provide the capabilities needed by them, in a defined environment and transform the stakeholder's view into a technical vision that meets the operational needs of users.

To fulfill this purpose and based on the good practices of RE identified as part of the construction of the theoretical framework, specific requirements divided by three MCDAI's maturity levels (basic, intermediate and advanced) were proposed (see Table 3¹). Requirements division in maturity levels was made to facilitate the model adoption through process improvement stepwise with small changes.

Table 3. Specific requirements of the RE base process.

Basic level
RE 1 Define the relevant stakeholder requirements. (1 and 9)
RE 2 Analyze and specify the requirements. (2, 6 and 7)
RE 2.2 Prioritize requirements. (7)

¹ In table 3, you can find the requirements statements distributed by levels, and in parentheses, it is related to the good practices identified in table 1.

RE 3 Achieve understanding and commitment to technical requirements. (10 and 11)
RE 4 Validate technical requirements. (8)
CM 4 Control changes. (12)
Intermediate level
RE 5 Model the technical requirements. (3 and 5)
Advanced level
RE 2.1 Approve technical requirements.
RE 5.1 Modeling requirements based on reuse.
RE 6 Establish bidirectional traceability. (13)
QA 6 Perform inconsistency reviews. (14)

RE 1 Define the relevant stakeholder requirements.

The appropriate sources and suppliers shall be identified to obtain relevant stakeholder requirements. The requirements shall be defined based on the needs and expectations of the suppliers and an analysis of the sources identified.

Recommended evidence: Providers list and Requirements list.

RE 2 Analyze and specify the requirements.

The stakeholder requirements shall be analyzed taking into account whether they are necessary or sufficient to meet the objectives of the product; from this analysis, new derived and/or implicit requirements can be defined. The functional and non-functional requirements shall be formally specified and with sufficient technical detail. Shall be reviewed the viability of technical requirements.

Recommended evidence: Requirements specification.

RE 2.1 Approve technical requirements.

A benchmarking shall be carried out in the corresponding application domain, to identify functionalities of similar products. The functionalities identified with the technical requirements shall be homologated, and define additional requirements that the product could contain to increase customer satisfaction.

Recommended evidence: Requirements specification.

RE 2.2 Prioritize requirements.

Priority to requirements that will be implemented according to the stakeholder needs, market conditions, and/or business objectives, shall be given.

Recommended evidence: Prioritization of requirements.

RE 3 Achieve understanding and commitment to technical requirements.

Shall be achieved requirements understanding between the suppliers and the project team. Shall be resolved conflicts arising between the requirements. Shall be obtained the project team commitment with the current and approved requirements implementation, as well as making the necessary changes, to plans, activities, and related work if the requirements evolve.

Recommended evidence: Tasks in the management tool (assigned and accepted), Meeting notes.

RE 4 Validate technical requirements.

The technical requirements shall be validated to ensure that the resulting product meets the stakeholder needs and

expectations and works as intended, in the environment of the end user.

Recommended evidence: Requirements specification.

RE 5 Model the technical requirements.

Shall be modeled the technical requirements to obtain a better understanding of the product to be developed. Shall be grouped the requirements taking into account criteria.

Note: The modeling of the requirements could be done taking into account different paradigms such as Structured Analysis; Object-Oriented Analysis; among others.

In the first case, models are created to represent the flow and content of the information (data and control), the product is divided into functional and behavioral partitions and the essence of what is to be built is described. For example: Data Flow Diagrams (DFDs); State Transition Diagram (DTE); Data Dictionary. In the second case, the objective is to model the concepts (objects) of the domain of the product, its relationships and behaviors. That model is continuously refined until a model with sufficient detail is obtained for its implementation in the form of executable code. For example: Use Case Models and Operation Scenarios; Class Model; Sequence and activity diagrams; State diagrams.

Recommended evidence: Document realization of requirements.

RE 5.1 Model requirements based on reuse.

A domain model(s) shall be defined and maintained that describes the borders of each domain with reuse potential and specifies its characteristics, capabilities, common elements and variants, optional or mandatory. The domain model(s) shall be incorporated into a repository of reusable assets, once they are formally evaluated and approved.

Recommended evidence: Domain model.

RE 6 Establish bidirectional traceability.

Bidirectional traceability between the project's objectives, stakeholder requirements, technical requirements, derived work products, and tasks that will fulfill it, shall be determined. Shall be updated traceability throughout the project as appropriate.

Recommended evidence: Traceability tool with the built-in elements.

To obtain the desired capacity level of the RE base process, in addition to fulfilling the specific requirements described above, the following shall be met:

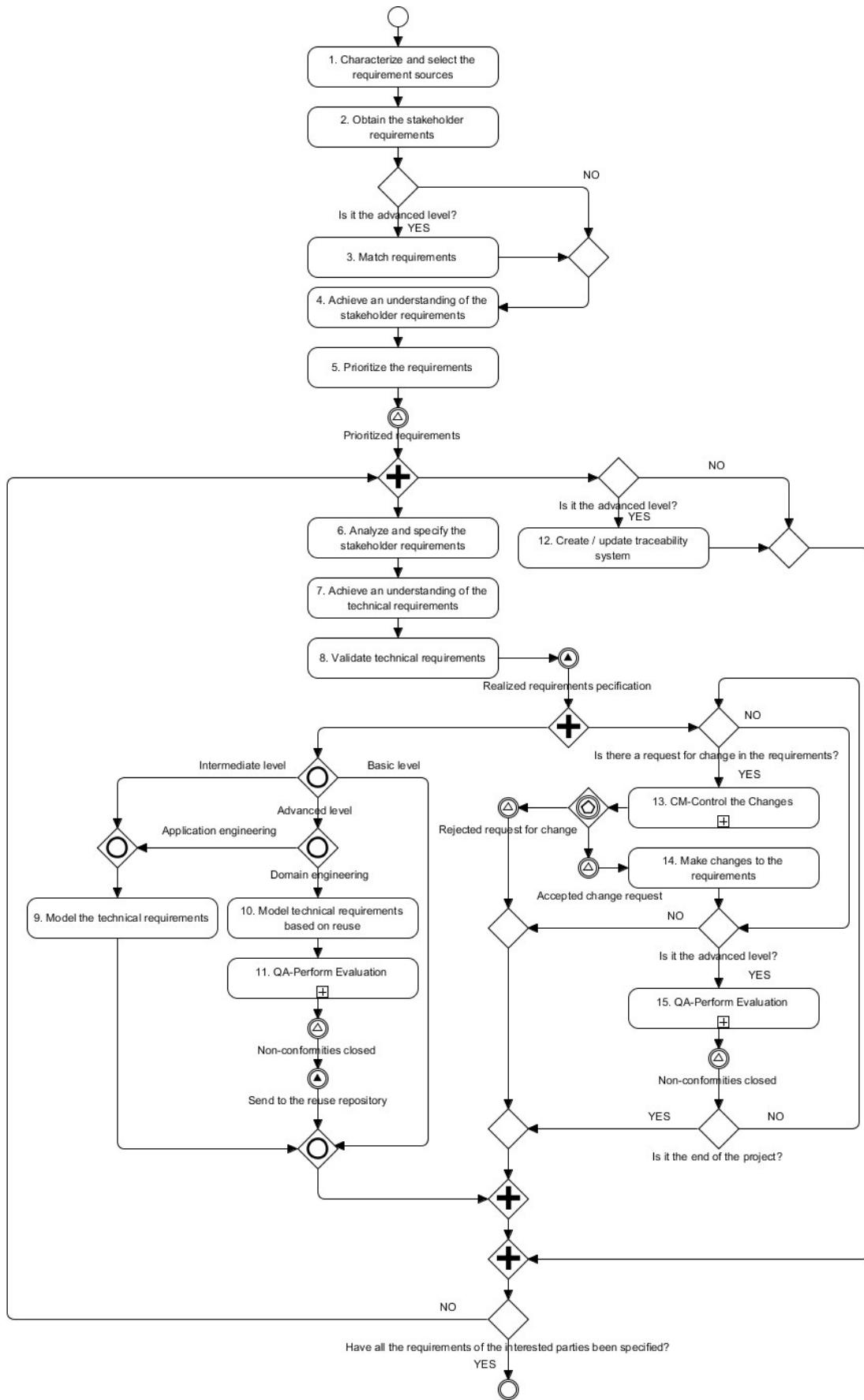
- For Basic level with the specific requirement, **CM 4 Control changes**, from Configuration Management base process, to manage requested changes on requirements.
- For Advanced level with the specific requirement, **QA 6 Perform inconsistency reviews**, from Quality Assurance base process, to ensure alignment between project work and requirements.

For the construction of the specific requirements described above were executed three stages. First, the authors prepared a proposal taking into account their experience and the good practices identified in Table 1. Second, was presented the proposal to 22 researchers who were working on the definition of the MCDAI, for dividing the specific requirements into the three maturity levels (Basic, Intermediate, and Advanced) of the model, and for identifying the relationship with the rest of the MCDAI's base processes.

The third stage executed after updating the proposal with the obtained feedback. Seven experts were identified, with an average of 7 years of experience working on the RE discipline, 100% engineers in computer science and with the scientific category of master. The specific requirements and the proposal of at what level they might be grouped were presented to experts, to obtain their assessment of them. The feedback with the experts allowed updating the specific requirements and the levels that group them. Finally, the last version of the RE base process, shown in the next section, was obtained.

3.2. Process and activities

As part of the solution, the graphic description (see Figure 1) and the textual description of the RE base process as an example of how to put the specific requirements into practice, is proposed.

**Figure 4.** Graphic description of the RE process.

Below is the textual RE process description.

1. Characterize and select the requirement sources.

The Analyst and the Client taking into account the stakeholders identified in PPMC - Project Planning, Monitoring, and Control (Batista Anisbert Suárez et al., 2016), obtain requirements source and characterize them. For the Advanced level when domain engineering is applied, the development is directed to an application family, therefore, requirement sources vary from specific clients to market and business studies; when domain engineering is applied, the sources are given by domain assets and the specific client.

The Analyst, Project Manager, and Client select requirement sources taking into account their characterization and the provider(s) that represent the client's interests, if applicable, and take responsibility for providing the requirements.

The "Requirement sources list" is obtained as a result of the execution of this activity.

2. Obtain the stakeholder requirements.

The Analyst uses the "Requirement sources list", the "Offer", and/or the "Technical Project" prepared when the project was conceived to analyze the requirements that would be needed to comply with the project's goals. Also, he identifies providers' needs and expectations, characterizes the organization operating environments, and prepares a comprehensive list of them. This list is continuously updated by monitoring any changes that may occur going forward and based on suppliers' suggestions. In case of not being satisfied, the analyst re-identifies/improves the requirements with the help of other techniques such as prototype, focus group, business use cases, business process model, among others.

For the Advanced level, when domain engineering is applied, the requirements are obtained through market and business studies and the analysis of past projects. In these cases, usually, the project does not have a specific client since it is working in the development of a generic product, for this reason, the analysis to resolve conflicts between requirements is made with functional experts. When application engineering is applied, the obtaining of the requirements is done by analyzing the existing domain assets with the specific client, for which common and variational elements, optional or mandatory, are taken into account, if necessary, to adopt them or design new requirements.

The "Requirement and restrictions list" is obtained as a result of the execution of this activity.

3. Match requirements.

From the Advanced level, the Analyst taking into account the "Requirement and restrictions list" performs a benchmarking in the corresponding application domain to identify similar products. Also, he matches the stakeholder's requirements with similar product functionalities to identify additional requirements and verify that the identified functional needs correspond to this product type.

The "Benchmarking" and "Requirement and restrictions list" with new requirements in cases to apply is obtained as a result of the execution of this activity.

4. Achieve an understanding of the stakeholder requirements.

The Analyst, taking into account the "Requirement and restrictions list", identifies the conflicts between the requirements and makes proposals on how to eliminate them, using functional experts.

The Analyst and Stakeholders meet to reach a consensus on the resolution of the conflicts identified, taking into account the proposal made.

The Project Team and the Requirements Providers achieve a common understanding of the "Requirement and restrictions list".

The "Requirement and restrictions list" (Updated) is obtained as a result of the execution of this activity.

5. Prioritize the requirements.

The Analyst taking into account the "Requirement and restrictions list" identifies the appropriate method for prioritization of the stakeholder requirements (e.g., hierarchical analysis, cumulative vote, numerical assignment, value-based prioritization, cost, and risks, among others). Also, he prioritizes the requirements using the appropriate method.

The "Requirement and restrictions list" with the prioritized requirements is obtained as a result of the execution of this activity.

6. Analyze and specify the stakeholder requirements.

The Analyst taking into account the "Requirement and restrictions list" with the prioritization groups the functional and non-functional requirements that correspond to the iteration. Analyzes if he can reuse the requirements of the previous projects. Identifies whether the requirements are necessary or sufficient to develop the product that satisfies the stakeholder and, if required, identifies new derived requirements (functional requirements). He refines the functional requirements in terms of its description and functionality details. He analyzes the "Requirement and restrictions list" taking into account the software product quality model defined in NC-ISO/IEC 25010, to identify implicit requirements (non-functional requirements). He refines the non-functional requirements by assigning allowable values to the quality attributes that the product should have. He reviews functional and non-functional requirements viability to determine if they are complete, feasible, and verifiable. From the Intermediate level, he also specifies the internal and external interface requirements of the system.

The "Requirements specification" is obtained as a result of the execution of this activity; hereafter these requirements will be treated as technical requirements.

7. Achieve an understanding of the technical requirements.

The Analyst and Stakeholders taking into account the "Requirements specification" meet to arrives at a common understanding of described technical requirements.

The “Requirements specification” (Updated) is obtained as a result of the execution of this activity.

8. Validate technical requirements.

The Project Manager, Analyst, and Client taking into account the “Requirements specification” validate the technical requirements using the prototype technique, where candidates of the system interfaces and the input and output elements are shown to the final user.

The Analyst, in case of any indication or observation by the clients, updates the “Requirements specification”.

The “Requirements Specification” (Signed) is obtained as a result of the execution of this activity.

9. Model the technical requirements.

From the Intermediate level, the Analyst defines the conceptual model, establishing the relationship of the entities of the system or subsystem and their fundamental attributes, future persistent classes, and candidates for the data model. Also, he models the requirements by making a technical description (Use Case Model, Operation Scenarios, Class Model, and User Stories, where applicable).

The Project Manager and Architect distribute the requirements in the modules or subsystems of the project.

From the Advanced level, the Domain Model corresponding to the applications family is taken into account to define the Product Analysis Model that is going to be developed for a specific client.

The “Analysis model” is obtained as a result of the execution of this activity.

10. Model technical requirements based on reuse.

From the Advanced level, the Analyst defines the domain model where are described its boundaries with other domains, and specifies the characteristics, capacities, common elements, and variants, optional or mandatory. He defines the conceptual model, establishing the relationships of the entities that are part of the domain and its fundamental attributes, future persistent classes, and candidates for the standard data model for the family of applications. Also, he models the requirements by making a technical description of the applications family (Use Case Model, Operation Scenarios, Class Model, and User Stories, in the cases that apply).

The Project Manager and Architect distribute the requirements in the project modules or subsystems.

The “Domain Model” as a result of the execution of this activity, is obtained.

11. QA-Perform Evaluation.

The Evaluation Team verifies that the “Domain Model” is technically correct guided by the sub-process QA-Perform Evaluation (Y. A. Lazo, 2016).

The “Evaluation file” is obtained as a result of the execution of this activity.

12. Create/update traceability system.

In the Advanced level, the Analyst taking into account the Traceability Guide inserts in the selected tool, as they are being developed the objectives of the project, the stakeholder requirements, the technical requirements, the work products and the tasks that they will comply with the agreed requirements. He establishes the corresponding bidirectional relationships between the elements inserted in the tool. Also, he updates the tool, if changes to the requirements or work products arise.

The traceability tool, with its established relationships is obtained as a result of the execution of this activity.

13. CM-Control the Changes.

The Change Control Committee analyzes the change request on the requirements as established by the subprocess CM-Control the Changes (García, 2017).

The “Change request” (accepted) is obtained as a result of the execution of this activity.

14. Make changes to requirements.

The Analyst taking into account the “Change request” accepted, at the Basic and Intermediate levels, makes the corresponding changes on the requirements and the related work products. In the Advanced level, the changes are made using the traceability tool.

The “Requirements specification” and related (updated) artifacts are obtained as a result of the execution of this activity.

15. QA-Perform Evaluation.

At the Advanced level, the Evaluation Team executes the inconsistency review between the requirements and the associated work products, taking into account the tool and the traceability guide, as established by the QA-Perform Evaluation sub-process (Y. A. Lazo, 2016).

The “File of the evaluation” is obtained as a result of the execution of this activity.

3.3. RE base process relationship with MCDAI

The RE base process has a close relationship with other base processes that compose the MCDAI. This relationship allows providing input elements for other base processes (see **Figure 5**). For example, the requirements specification and the domain model are input elements of the TSD base process and are taken into account for product design and implementation. In this relationship, it is also appreciated that the result of other base processes is used in the RE base process, for example, the changes requests to requirements are accepted or rejected by the CM base process, among others.

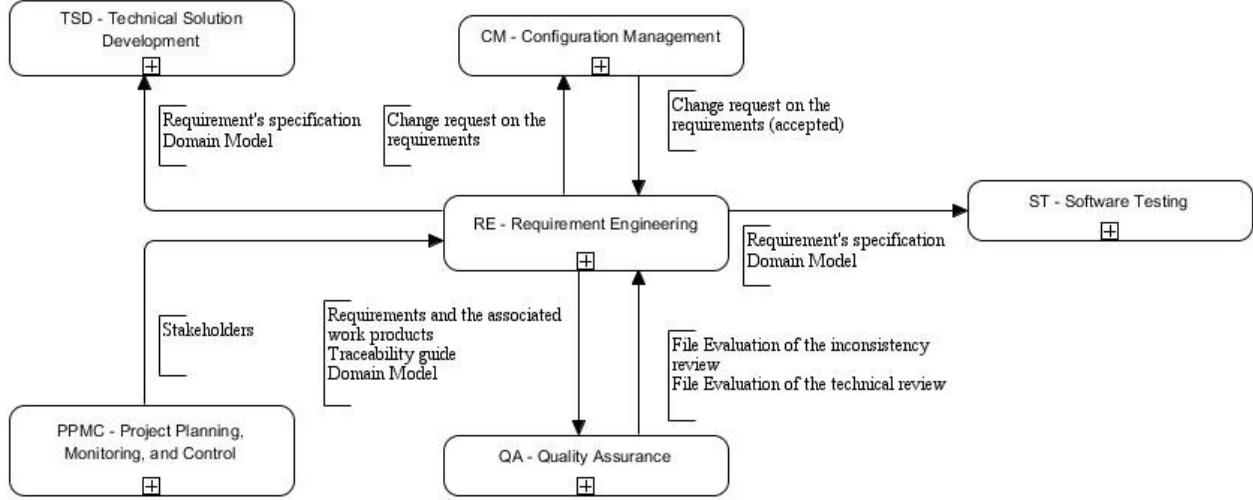


Figure 5. Relationship of the RE base process with other MCDAI processes.

Also, this relationship assures to comply with the model's generic requirements. As shown in **Figure 6**, through the OPM base process the RE process is defined, is defined its associated roles and responsibilities, is provided the resources necessary to execute the process, and is institutionalized the RE process throughout the organization. PPMC base process, plan and monitor the RE process execution, as well as manage the training of the project personnel internally to the process execute. CM base process identifies and preserves the configuration

elements that are generated in the RE process. QA base process evaluates that the defined RE process is executing in the organization, and keep the management informed of the status of that process. MI base process defines necessary indicators to measure the RE base process and makes improvements to it. KM base process manages the project team training about the RE process, that could not be satisfied in the project, and knowledge generated by it. Finally, the RM base process identifies and treats risks associated with the RE base process.

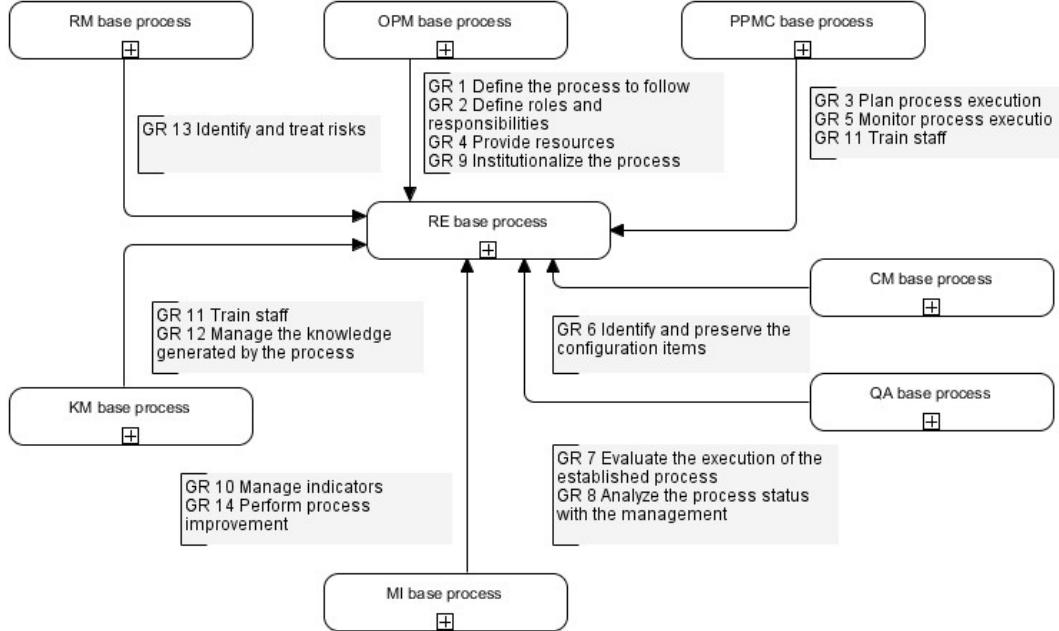


Figure 6. RE base process compliance with generic requirement.

3.4. Measure the RE base process

To measure the RE base process influence on the software development projects' success is proposed the indicator Requirement Compliance Index (RCI). It aims to evaluate requirements compliance agreed with clients. It as an unfulfilled requirement is understood when it has not been developed, or are not those agreed-upon results

obtained after its implementation. For this, are identified the following base measures (ARQ: Agreed requirements quantity, RIQ: Requirements implemented quantity). The following measurement function $RCI = \frac{ARQ}{RIQ}$ is used to calculate the RCI. The projects are considered successful if $RCI > 0,95$. This indicator was selected by 7 experts with

an average of 7 years of experience working in the discipline of requirements engineering.

4. Validation

4.1. Analysis of the proposal taking into account focal group

The authors of the present investigation consider that the focal groups constitute a valuable and widely used technique to obtain information. For this reason, they decided to use it to know if the solution proposal uses the correct terminology and is technically viable. For its conformation the criteria issued by Aigneran and Méndez were taken into account (Aigneran, 2009; Méndez, 2007), those who state that the size of the group should oscillate between 4 and 12 participants; that all the participants have the possibility of issuing their criteria; and that the group must be homogeneous in order to ensure the diversity of ideas. In order to comply with the above, 12 specialists were summoned, with more than 5 years of experience in the roles of analyst and architect. The selected ones represented the organizations CALISOFT, DESOFT, XETID, ETECSA, TRANSOFT, EICMA, AICROS and SEGURMÁTICA (A. Y. Lazo, Tamayo, Enamorado, Pérez, & Sánchez Osorio, 2018). The final result was a RE process enriched with the experiences of each participant and the unanimous criterion that it is an accepted proposal that meets the needs of the software development organizations in Cuba.

4.2. Analysis of the implementation of the process in pilot projects

A pre-experiment was applied in pilot projects to evaluate that when introducing the RE base process in software development projects, the project's success is greater than 48%, concerning the dimension of the Requirement Compliance Index variable.

Sampieri suggests pre-experiment can be done, through a case study with a single measurement, or the design of a pre-test - post-test with a single group (Hernández, Fernández, & Baptista, 1991). When analyzing the two options, it was found that in the first one, there is no manipulation of the independent variables, and there is no previous reference of what was the situation before performing the stimulus. In the second one, there is an initial reference point to see what level the group had in the dependent variables before the stimulus, allowing a follow-up.

Taking into account the foregoing, researchers selected the second variant, knowing that pre-experimental designs are not suitable for the establishment of relationships between independent and dependent variables. But they consider it important because can yield results that when compared with those of other methods help to reach conclusions.

To implement the pre-experiment in the period from January 2018 to July 2018 were selected six projects from 3

different organizations (DATYS, AICROS, and TRANSOFT). The projects were developing web applications, with a team of six people, with mastery of the technologies used and with more than 5 years of experience in the business, the requirements average was 110, functional and non-functional.

At the end of the pre-experiment, five of six pilot projects were successful taking into account the RCI indicator, which represents 83.33%. According to **Table 4**, project 2 was the one that did not reach an $RCI > 0.95$.

Table 4. RCI of project.

Projects	RIQ	ARQ	RCI
P1	120	120	1.00
P2	108	120	0.90
P3	99	100	0.99
P4	110	110	1.00
P5	110	110	1.00
P6	97	100	0.97

Carrying out a comparative analysis between the diagnosis made in 2017 and the result obtained, in the first case only 48% of the projects completed successfully, and in the second case, an improvement in the indicator is seen in 83.33%. However, carrying out an exhaustive analysis of the project that did not comply with the indicator, in the review of adherence to the RE process, it reached a 50% implementation of the activities, an aspect that could influence the obtained results.

Among the RE process activities not executed in some project iterations, were analysis, negotiation, and validation of the requirements, due to the distance between the client and the project team. The absence of these activities caused that there was no understanding between the parties about the requirements in early stages and that the client was dissatisfied with seven of the agreed requirements because they did not work as expected and another five had problems related to usability.

The results allowed the authors of the research to appreciate an improvement in the success of the projects taking into account the dimension of the RCI variable, after introducing the proposed process.

4.3. Satisfaction of end users

The technique of V.A. Iadov was created by N.V. Kuzmina in 1970, for the study of satisfaction with pedagogical careers. Subsequently, it has been used in several investigations to evaluate satisfaction in different contexts. Iadov consists of five questions, three closed, and two open. In this research, this technique is used to assess user satisfaction in pilot projects respect to the RE process. For this, a survey was applied to six analysts and six architects. The criteria measured in the survey are based on the relationships established between the three closed questions, related through the Iadov Logical Table (see **Table 5**).

Table 5. Iadov Logical Table for the RE base process (modified by the authors of this research).

	1. Do you consider the Requirements Engineering Base Process complex and difficult to understand?								
	No		I don't know			Yes			
3. Is the Requirements Engineering Base Process used to your liking?	2. If you were to carry out another project, would you use the proposed requirements engineering process?								
	Yes	I don't know	No	Yes	I don't know	No	Yes	I don't know	No
Clearly pleased	1	2	6	2	2	6	6	6	6
More pleased than unpleased	2	2	3	2	3	3	6	3	6
Not defined	3	3	3	3	3	3	3	3	3
More unpleased than pleased	6	3	6	3	4	4	3	4	4
Clearly unpleased	6	6	6	6	4	4	6	4	5
Contradictory	2	3	6	3	3	3	6	3	4

The number resulting from the interrelation of the three questions indicates the position of each respondent on the satisfaction scale. Respondents used the following satisfaction scale, to which a value is assigned to determine the group satisfaction index:

1. Clearly pleased: +1
2. More pleased than unpleased: +0.5
3. Not defined: 0
4. More unpleased than pleased: -0.5
5. Clearly unpleased: -1
6. Contradictory: 0

Below is the calculation of the Group Satisfaction Index (GSI) in the following formula:

$$\begin{aligned} GSI &= \frac{A(+1) + B(+2) + C(0) + D(-0.5) + E(-1)}{N} \\ &= \frac{12(+1) + 0(+2) + 0(0) + 0(-0.5) + 0(-1)}{12} = 1 \end{aligned}$$

The group index yields values between + 1 and - 1 and is classified as follows:

- Satisfaction: Values between 0.5 and 1
- Contradiction: Values between -0.49 and 0.49
- Dissatisfaction: Values between -1 and 0.5

The result of $GSI = 1$, means maximum satisfaction for the proposed ER base process. This result was corroborated with the answers to open questions 4 and 5, where respondents expressed that they would not change anything in the base process because it fits their needs.

5. Conclusion

The good practices for RE were grouped, into requirements development and requirements management. Requirements development's main practices are identifying stakeholder needs, and specifying, analyzing and negotiating requirements. Requirements management's

main practices are controlling changes and maintaining traceability.

The graphic and textual description of the RE base process is a guide to adopt the MCDAI's requirements divided by the three levels of maturity and facilitate their adoption.

Incorporating feedback activities with clients in the RE process is a factor that influences the success of the project, because it allows identifying the necessary changes to the requirements, at the appropriate time, for the product to respond to the client's needs and expectations.

The proposal validation contributed to verify the user satisfaction with the proposed process and that the execution of the process can contribute to the project success.

It is recommended to measure the process impact on the volatility of the requirements to contribute to the project planning fulfillment.

References

- Aigner, M. (2009). La técnica de recolección de información mediante grupos focales. *La Sociología en sus escenarios*.
- Bastarrica, C. (2011). Productividad en la Industria TIC. *Bits*.
- Brun, R. E. (2007). *Técnicas de análisis de dominio: organización del conocimiento para la construcción de sistemas software*. Paper presented at the La interdisciplinariedad y la transdisciplinariedad en la organización del conocimiento científico: Interdisciplinarity and transdisciplinarity in the organization of scientific knowledge: Actas del VIII Congreso ISKO-España, León, 18, 19 y 20 de Abril de 2007.
- CALISOFT, C. N. d. C. d. S. (2014). *CS-03-D (14-001) Libro de diagnóstico*.
- CALISOFT, C. N. d. C. d. S. (2017). *CS-03-D (17-001) Libro de Diagnóstico*.
- CMMI Institute. (2015). Retrieved 02/11/2015, 2015, from https://sas.cmmiinstitute.com/pars/pars_detail.aspx?a=25323
- COMPETISOFT, P. (2006). COMPETISOFT-Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica. Versión 0.2. *Diciembre*.

- del Toro, A. A. (2018). *Una mirada desde el desarrollo ágil a los Requisitos de Software. Experiencias en Datys Villa Clara.* Paper presented at the Taller 2 - Ingeniería de Requisitos.
- García, Y. G. (2017). *Proceso Base Gestión de la Configuración para un Modelo de Calidad en Cuba.* Universidad de las Ciencias Informáticas.
- Goguen, J. A. (1994). *Requirements engineering as the reconciliation of social and technical issues* (San Diego: Academic Press Professional ed.).
- Hernández, S. R., Fernández, C. C., & Baptista, L. P. (1991). *Metodología de la investigación.*
- IEEE. (2014). *SWEBOK. Guide to the Software Engineering Body of Knowledge* (Versión 3 ed.).
- International, T. S. G. (2018). Chaos Report.
- ISO, IEC, & IEEE. (2015). ISO/IEC/IEEE 15288 Systems and software engineering — System life cycle processes.
- ISO, IEC, & IEEE. (2017). ISO/IEC/IEEE 12207, Systems and software engineering — Software life cycle processes.
- ISO, IEC, & IEEE. (2018). ISO/IEC/IEEE 90003, Software engineering — Guidelines for the application of ISO 9001:2015 to computer software.
- Lazo, A. Y., Tamayo, O. L., Enamorado, P. O., Pérez, M. D., & Sánchez Osorio, Y. (2018). *Apuntes sobre el Modelo de la Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI).* Paper presented at the XVII Convención y Feria Internacional Informática 2018, La Habana. <http://www.informaticahabana.cu/es/node/3703>
- Lazo, Y. A. (2016). *Proceso Base de Aseguramiento de la Calidad para el Desarrollo de Software en Cuba.* Universidad de las Ciencias Informáticas.
- Lehtinen, T. O., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures—An analysis of their relationships. *Information and Software Technology*, 56(6), 623–643.
- Losavio, F., Guzmán, J. C., & Matteo, A. (2011). Correspondencia Semántica entre los lenguajes BPMN y GRL. *Enl@ce*, 8(1).
- MANENE, L. M. (2013). Los diagramas de flujo: su definición, objetivo, ventajas, elaboración, fases, reglas y ejemplos de aplicaciones. *Los diagramas de flujo.*
- Manso Martínez, M., & García Peñalvo, F. J. (2013). Medición en la Reutilización Orientada a Objetos.
- McLeod, L., & MacDonell, S. G. (2011). Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR)*, 43(4), 24.
- Medina, Y. T. (2012). Modelado de procesos con IDEF en la metodología RUP. *Serie Científica-Universidad de las Ciencias Informáticas*, 5(2).
- Méndez, A. L. d. (2007). *La entrevista y los grupos focales.*
- Montoni, M. A., Rocha, A. R., & Weber, K. C. (2009). MPS. BR: a successful program for software process improvement in Brazil. *Software Process: Improvement and Practice*, 14(5), 289–300.
- Murcia-Oeste-Arrixaca, Á. I. (2013). Manual para el diseño de procesos.
- Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., . . . Little, R. (2007). A framework for software product line practice, version 5.0. *SEI-2007-<http://www.sei.cmu.edu/productlines/index.html>.*
- Oficina Nacional de Normalización. (2015a). NC-ISO 9001 Sistema de Gestión de la Calidad — Requisitos.
- Oficina Nacional de Normalización. (2015b). NC ISO 9000 Sistema de Gestión de la Calidad - Fundamentos y Vocabulario.
- Oktaba, H. (2005). Modelo de Procesos para la Industria de Software-MoproSoft-Versión 1.3, Agosto de 2005: NMX-059/01-NYCE-2005.
- Oktaba, H. (2015). Historia de una norma. MoproSoft y sus primeros pasos. Retrieved 1, 2015, from <http://sg.com.mx/content/view/390>
- Pérez, D. M. (2014). *Guía general para un Modelo Cubano de Desarrollo de Aplicaciones Informáticas.* Universidad de las Ciencias Informáticas. Retrieved from <https://repositorio.uci.cu/jspui/handle/ident/8725>
- Pérez, D. M., & Aveleira, D. Q. (2016). *Evolución del Modelo de la Calidad para el Desarrollo de Aplicaciones Informáticas.* Paper presented at the XVI Convención y Feria Internacional Informática 2016, La Habana. <http://www.informaticahabana.cu/es/node/664>
- Pressman, R. S. (2010). *Ingeniería de software. Un enfoque práctico* (Séptima edición ed.). México.
- Rosato, M. (2018). Go Small for Project Success. *PM World Journal*, VII(V).
- Salazar, L. L. (2017). *Desarrollo del proceso Solución Técnica para los proyectos de desarrollo de la Universidad de la Ciencias Informáticas.* Universidad de las Ciencias Informáticas (UCI).
- Silega, M. N. (2014). *Método para la Transformación Automatizada de Modelos de Procesos de Negocio a Modelos de Componentes para Sistemas de Gestión Empresarial.* Universidad de las Ciencias Informáticas (UCI).
- SOFTEX. (2009a). *MPS.BR - Mejora de Proceso del Software Brasileño* (Vol. Guía de Implementación – Parte 4: Fundamentos para Implementación del Nivel D del MR-MPS).
- SOFTEX. (2009b). *MPS.BR - Mejora de Proceso del Software Brasileño* (Vol. Guía de Implementación – Parte 1: Fundamentos para Implementación del Nivel G del MR-MPS).
- Sommerville, I. (2011). *Ingeniería de software* (Novena Edición ed.). México.
- Suárez, B. A. (2013). *Marco de procesos para las Entidades de Servicios de Tecnología de la Información de la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informáticas (UCI).
- Suárez, B. A., Sánchez, O. Y., Muñoz, R. M., Ruenes, C. S. B., Gómez, B. C., Gutierrez, F. L. M., & Calunga, Á. A. (2016). *Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas: categoría de Gestión de Proyecto.* Paper presented at the XVI Convención y Feria Internacional Informática 2016, La Habana.
- Team, C. P. (2010). CMMI® for Development, Version 1.3, Improving processes for developing better products and services. no. *CMU/SEI-2010-TR-033. Software Engineering Institute.*
- The Standish Group International. (2014). The Standish Group Report.
- The Standish Group International. (2015). Chaos Report 2015.