Quality Assurance methodology for system requirement definition

Karla Gomez Sotelo

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Institut National des Sciences Appliquées de Toulouse

---

Présentée et soutenue par

Karla GOMEZ SOTELO

Le 5 juillet 2019

Méthodologie d'assurance de la qualité pendant la définition des exigences d'un système

---

Ecole doctorale : **SYSTEMES**

Spécialité : **Génie Industriel**

Unité de recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
CLAUDE BARON et Philippe ESTEBAN

Jury

M. Marc ZOLGHADRI, Rapporteur
M. Eric BONJOUR, Rapporteur
M. Abd-El-Kader SAHRAOUI, Examinateur
Mme Lidia LOPEZ CUESTA, Examinatrice
Mme Isabel Maria PINTO RAMOS, Examinatrice
M. Rob VINGERHOEDS, Examinateur
Mme Claude BARON, Directrice de thèse
M. Philippe ESTEBAN, Co-directeur de thèse
Mme Citlalih Y. A. GUTIERREZ ESTRADA, Invitée
M. Pierre DE CHAZELLES, Invité

Acknowledgments

These three years have changed my life.

I will start by thanking my supervisors Professor Claude BARON and Professor Philippe ESTEBAN, who welcomed me into their research team, accompanied me throughout the process, and were patient with me, thank you. I continue to thank Professor Citlalih Y. A. GUTIERREZ ESTRADA because without her support and affection I could not even have started this adventure, she has helped me, unconditionally, until the end.

I thank my dear Mexico, which through the National Council for Science and Technology (CONACYT) provided me with the means to continue my studies. I am also grateful to the beautiful France, which welcomed me with open arms in the LAAS-CNRS facilities, providing me with the means to work every day.

I thank Regina and Sebastian, who have been courageous and hard-working, and Maru who at a distance is always taking care of us. I thank my friends: Diego, Xin, Cathy, Sophie, Anais, Chantal, Marshall, Juan, Julie, Suzie, Joana, Ombretta, Arturo, Luis, Abraham, Baptiste, Min, Guillaume, Vatsal, and Yassine, who have given me their friendship and now are my family.

I thank each member of the ISI team, professors and doctoral students, who shared this time with me, encouraged me, and always gave me a smile.

Thank you all.

Abstract

Quality Assurance methodology for system requirement definition

Companies are competing to put their products on the market. In this race, knowledge of the quality characteristics that end users require for the product is sometimes presupposed or misunderstood. The result is often a product that does not achieve the purpose for which it was designed and manufactured. In this context, is it possible to guide the development process methodologically in order to ensure the quality of a product? With reference to Systems Engineering, it is at the stage of Concept in the life cycle of the system that the needs of stakeholders are collected, translated first into stakeholder requirements and then into system requirements. This thesis therefore addresses these steps as a priority. It proposes a methodology to ensure that stakeholder needs are well understood and properly translated into system requirements. The proposal complies with the ISO 15288 (2015) quality standard and incorporates the Lean principles. The thesis also proposes a tool that supports the methodology. The results obtained from several case studies developed at the Tecnológico Nacional de México, Instituto Tecnológico de Toluca (ITTol), Mexico, demonstrate the effectiveness of the proposed methodology. Its use increases the likelihood that the delivered product will meet stakeholder expectations, reduces requirement changes due to misidentification of needs and, therefore, the costs incurred by these changes, and ensures faster delivery of the product to the market.

Keywords: Systems Engineering, Quality Assurance, system requirements, stakeholder needs, quality.

Résumé

Méthodologie d'assurance qualité pour la définition des exigences du système

Les entreprises sont en compétition pour mettre avant les autres leurs produits sur le marché. Dans cette course, la connaissance des caractéristiques de qualité qu'un utilisateur final souhaite pour le produit est parfois présupposée ou mal comprise. Il en résulte souvent un produit qui n'atteint pas l'objectif pour lequel il a été conçu et fabriqué. Dans ce contexte, est-il possible de guider méthodologiquement le processus de développement afin d’assurer la qualité d'un produit ? En nous référant à l'Ingénierie des Systèmes, c’est à l'étape de la définition du Concept dans le cycle de vie du système que les besoins des parties prenantes sont recueillis, traduits tout d’abord en exigences des parties prenantes puis en exigences sur le système. Cette thèse adresse donc prioritairement ces étapes. Elle propose une méthodologie visant à assurer que les besoins des parties prenantes sont bien compris et correctement traduits en exigences système. La proposition est conforme à la norme de qualité ISO 15288 (2015) et intègre les principes du Lean. La thèse propose également un outillage qui supporte la méthodologie. Les résultats obtenus sur plusieurs études de cas développés à Tecnológico Nacional de México, Instituto Tecnológico de Toluca (ITTol), Mexique, démontrent l’efficacité de la méthodologie proposée. Son utilisation augmente les chances que le produit livré réponde aux attentes des parties prenantes, réduit les changements d'exigences dus à une mauvaise identification des besoins et, par conséquent les coûts induits par ces changements, et assure une livraison plus rapide du produit sur le marché.

Mot clefs : Ingénierie des Systèmes, Assurance de la Qualité, exigences du système, besoins des parties prenantes, qualité.

Content

General Introduction

This chapter introduces the research work developed at the Laboratoire d'Analyse et d'Architecture des Systèmes of the Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse, France, in collaboration with Tecnológico Nacional de México, Instituto Tecnológico de Toluca (ITTol), Mexico, and with the research funding (scholarship) from the Mexican National Council for Science and Technology (CONACYT).

The research was carried out in the ISI Team (Ingénierie Système et Integration; Systems Engineering and Integration), that addresses the design of complex system design, considering methods, processes, and tools to that aim. That way, the ISI team conducts research works for the improvement of life-cycle processes like requirement engineering, design, verification & validation, modeling, and simulation.

The reader will find in this General Introduction the background and motivation to carry out this research work, followed by the research question, the general objective, and expected results; finally, the structure of the thesis.

Background and motivation

Companies compete to deliver products to market (Jakjoud, Zrikem, Baron & Ayadi, 2014; Kiritani & Ohashi, 2015), which forced them to optimize their design and manufacturing processes (Jakjoud, Zrikem, Baron & Ayadi, 2013).

Nevertheless, in this race to the market, the knowledge of quality characteristics that an end user wants to meet is sometimes pre-assumed or misunderstood. Faisandier (2014) specifies that there are not necessarily strong links between needs, their declination into technical requirements then potential solutions. It often results in a product that does not accomplish the purpose for which it was requested, designed and manufactured (Kiritani & Ohashi, 2015). Currently, literature describes "what" is the process of need identification and the translation of needs into requirements, but they do not say "how". The different standards and guides tells what to do, but not how. No information that explicitly shows *how* the *customer needs* are translated into *system requirements* is available, resulting in a poor quality of product development (Blanchard & Fabrycky, 2006; Gómez Saavedra, 1991; Gutierrez Pulido, 2010; Kiritani & Ohashi, 2015). Little by little, "*companies have realized that it is important to pay attention to how they can best design their products*" (Langen, 2002): to design the "how" is the opportunity of every company to make the difference compared with other organizations.

According to Badreau & Boulanger (2014), a *good* system meets its goals, justifies its reason for *being* in a rational way, is operational, and overcomes obstacles in its different phases of life.

Several authors (Atkinson, 1990; Blanchard & Fabrycky, 2006; Kiritani & Ohashi, 2015) recognize the need that a greater effort should be done in the initial system requirement definition, in order to ensure effectiveness during design process decision making. Kritani & Ohashi (2015), specify "*the quality of requirements definition leads directly to the final success or failure of system development project*".

In accordance to Oehmen (2012), the topic "*unstable, unclear and incomplete requirements*" occupies the second place among the "*themes of challenges in managing engineering programs*", after "*firefighting - reactive program execution*", because it affects seriously the program's efficiency and effectiveness. Some examples of issues about "*unstable, unclear and incomplete requirements*" are: incomplete or erroneous understanding of stakeholders requirements, derived requirements are not identified, requirements are not formulated properly, requirements are possibly in conflict with one another, and an unclear understanding of stakeholders' perceptions of value. Badreau & Boulanger (2014) adds that the main causes of project failure related to requirements are: requirements are incomplete, users are not involved in the project, expectations are unrealistic, and inefficiency of requirement management when requirements evolve over the course of the project.

In addition, authors like Walton (1999) express that *learned lessons* for each new product development should be captured, documented and applied to future product developments, but there is not a methodology that serves as a guide to do it. Authors like Baines et al. (2006) add that the activities of the product design process that truly add value are an issue not solved yet, but there is the intention to establish and apply specific lean techniques to succeed in lean product design.

Knowing all these facts, a research question arises and motivates this research work.

Research question

Is it possible to methodologically guide the processes of identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product?

General objective and expected results

The general objective of this research work is focused on the development of a methodology and its tools, based on Systems Engineering and in compliance with the quality standard ISO 15288 (2015), to ensure, as far as possible, that the stakeholder needs are understood, and later, are translated into system requirements, thus reducing risks of project abortion or delays, and adding value to the analysis and design process.

The expected results are:

1. A methodology and a tool to guide the analysis and design team in understanding the stakeholder needs and translating them into system requirements.
2. The validation of the proposition through a case study (or more), generating documentation that demonstrates the methodology's follow-up, step by step, from the translation of stakeholder needs into system requirements.

The objective can be achieved through the integration of different domains, techniques, and methodologies. The application of quality tools and methodologies will decrease the time in which a new product will be available for the user, ensuring its efficiency and effectiveness, making the organization more competitive (Bauch, 2004). Lean Thinking philosophy will make processes run like a clockwork (Oehmen 2012), with value-added and waste reduced. Quality Assurance will monitor the processes to error prevention and requirement fulfillment (ISO 9000, 2005); while Systems Engineering will deal with the product life cycle, from need identification to product retirement stages (Blanchard & Fabrycky, 2006; Ryan & Faulconbridge, 2016; Faisandier, 2014).

This research work has the originality of integrating these different domains to support organizations during the Concept stage of the system life cycle, focusing on the earliest and most crucial activities, stakeholder needs identification and system requirement definition, including the generation of documents -for quality management purposes- in compliance with the standard ISO 15288 (2015) and, at the same time, obtaining a product or service that fulfills stakeholder needs.

Structure of the thesis

The report is organized as follows (Figure 1).

The ***Context*** of this research work has been introduced in this General Introduction and will be developed in Chapter 1.

> **General Introduction** exposed the background and motivations to carry out this research work, expressed the research question, defined the general objective (*developing a methodology and tools helping understanding the stakeholder needs and translating them into system requirements*), indicated the expected results (*the methodology, tools, and validation through case studies*), and now introduces the report organization.
>
> **CHAPTER 1 Problem context and research focus,** focus on the Systems Engineering discipline and on the quality standards related to Systems Engineering, for quality managing purposes and product quality assurance. It develops the research question (translating *stakeholder needs* into *system requirements*) in this context, and positions the research at the Concept stage of the system life cycle. It also defines the boundaries of this thesis

The ***Literature Review*** is covered by Chapters 2 to 5.

> **CHAPTER 2 Requirements Engineering** defines what a *requirement* is and shares the state of the art regarding how to express a *good* requirement statement, its attributes, and classifications. It reviews different requirements engineering process models. The chapter also highlights the importance of requirement traceability and its management through the system life cycle.

As the first expected result of this research work is to *understand the stakeholder needs and correctly translate them into system requirements*, the following chapters specifically address this issue. The translation of stakeholder needs into system requirements can be done in two successive stages, first, *translate stakeholder needs into stakeholder requirements*, then *translate stakeholder requirements into system requirements*. Each stage will benefit from a dedicated focus in respective chapters 3 and 4. It is important to point out that this research work is based on the bibliographic revision of complex systems in general, independently of the type of systems or application domain.

> **CHAPTER 3 From Stakeholder Needs to Stakeholder Requirements** reviews the literature on the translation of stakeholder needs into stakeholder requirements process. It identifies and describes its principal activities and tasks.
>
> **CHAPTER 4 From Stakeholder Requirements to System Requirements** describes the state of the art to translate stakeholder requirements into system requirements. It collects and compares the different points of view of several authors and how they propose to carry out the activities involved in this process.

Literature is substantial regarding on how to conduct these two processes, their activities, and related tasks. We took the opportunity to analyze and integrate all relevant contributions into one well-defined process to help to solve the problem. Our first conclusion on this analysis is the following: in spite of the numerous contributions found in literature, we could not find any clue on how to perform some activities or tasks. For instance, there is not any objective method to *weigh* the stakeholder level of importance, nor any structured method that helps to elicit needs, nor even a clear definition of what *bi-directional traceability at adjacent levels of the system* is. This will thus be a point we will address in our research.

Additionally, there is still an opportunity to design a lean process to translate *stakeholder needs* into *system requirements* that adds value while waste is reduced. For that reason, we decided to have a look at Lean Thinking, because this philosophy considers value, waste, and create value while reducing waste as main fundamentals, in order to address the sub-objective *reduce the risks of project abortion or delays, and add value to the analysis and design process*.

> **CHAPTER 5 Lean Thinking** enounces the fundamental concepts of *Lean Thinking*, how this philosophy has been applied to enable *lean* into the system life cycle phases (Systems Engineering), and how *lean* is being implemented in quality management systems (Quality Assurance).

At this point, considering the research objectives, the lessons learned from the state-of-the-art analysis and our conclusions from these analyses, Chapter 6 come to some proposals of contributions.

> **CHAPTER 6 DREAM: a Quality Assurance Methodology for System Requirements definition.** This chapter presents the research methodology we adopted to develop DREAM, and answer the research objective. We called this methodology *DREAM*, for *Driven Requirements Analysis Management*. It focuses on supporting the two stages of translation of stakeholders needs into system requirements, *from Stakeholder Needs to Stakeholder Requirements*, and *from Stakeholder Requirements to System Requirements*, as recommended by Systems Engineering
>
> In addition, Chapter 6 presents *DREAM* as a *quality management system.* The chapter presents the different *DREAM processes*, *work instructions*, and *forms* as a complete package to, prior training, be implemented into an organization.

Once *DREAM* is elaborated, it is necessary to verify if it answers the research question: *Is it possible to methodologically guide the processes of identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product?*

To verify if *DREAM* achieves the goal of providing a *methodology and tools, based on Systems Engineering, compliant with the quality standard ISO 15288 (2015), to ensure that the stakeholder needs are understood and later translated into system requirements.* That is the goal of Chapter 7.

> **CHAPTER 7 DREAM Verification & Validation** reveals the results of verification and validation of *DREAM* as a *quality management system*. The verification is carried out through comparing *DREAM* against *ISO 15288 (2015)* for quality assurance purposes of the system. Due to the nature of the research, two qualitative methodologies are applied to validate DREAM: a) *Case Study* for the application of DREAM as a *quality management system*; and b) *Questionnaires* to obtain data, evaluate the proposal, and subsequently improve it.

The results confirm that *DREAM* complies with the standard ISO 15288 (2015).

Moreover, *DREAM* is validated by professors and students that implemented it in student projects; their commitment and help is invaluable because the students, who did not have any expertise in conceptualizing new systems, found *DREAM* useful as a *methodological guide for identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product.*

Nevertheless, students found that *DREAM* can sometimes be quite difficult to apply at some stages. They suggested that it would be particularly useful if the process was supported by an information technology-based tool that would accompany the process, reduce document management, decrease the complexity of the forms proposed, and save time and effort; also, they also expressed their interest in having a prior training before applying *DREAM.*

In conclusion, the research question is answered, and the expected results have been finally achieved:

DREAM and its tools ensure, as far as possible, that the analysis and design team has understood the stakeholder needs and has translated them into system requirements.

The proposition was validated through several *student* case studies, generating documentation that demonstrates the methodology's follow-up, step by step, from the translation of stakeholder needs into system requirements.

Conclusions & Perspectives. This chapter synthesizes the research work, from the initial problem to the results obtained through the proposal and opportunities for improvement. Some perspectives are indicated for future work.

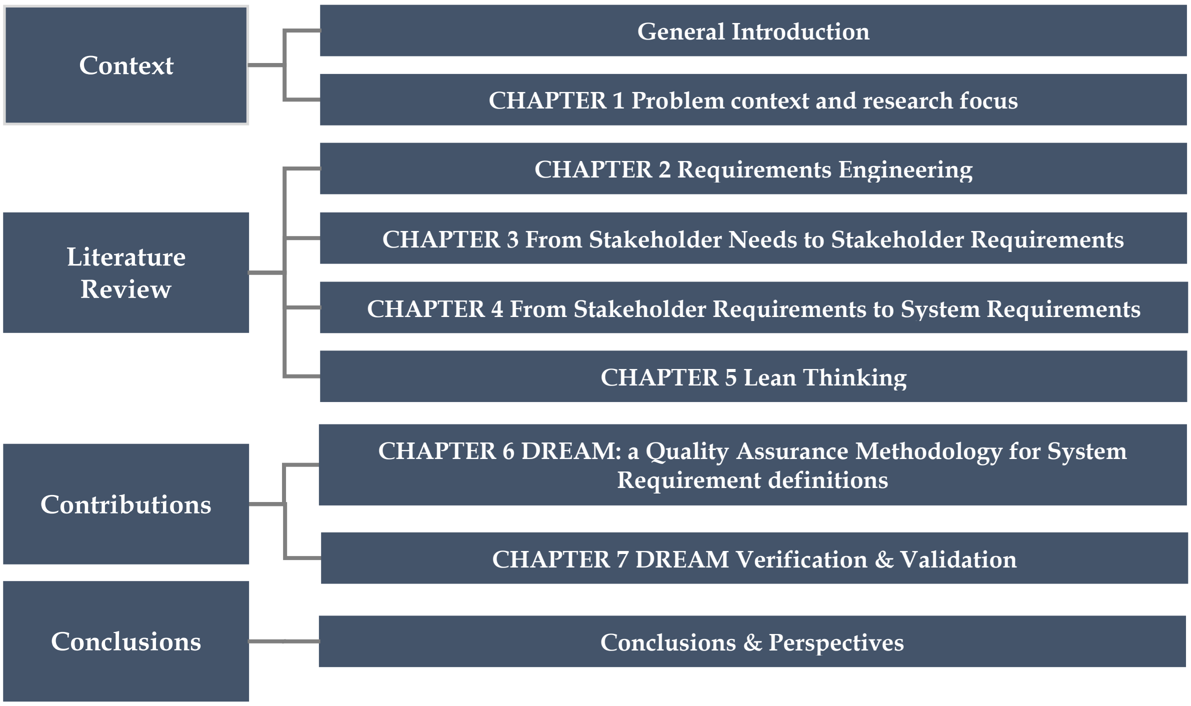


Figure 1 Structure of the thesis

CHAPTER 1 Problem context and research focus

1.1 Introduction

In general, Engineering Science has the objective to improve people life quality through problem solution. Acosta Flores (2002) state that technical knowledge applied by engineering allow increasing the satisfaction of social needs as a whole.

But the problems society faces every day are numerous and they are interrelated, it is a complex problematic situation without an easy nor unique solution; among the several alternatives, one solution could be better in a specific context, but it could be overpassed by other alternatives in different contexts. And in this point is where Systems Engineering (SE) is considered *valuable*, because of its wide vision of the problem not only in an isolated part of it (Acosta Flores, 2002).

In this chapter, the reader will find the meaning of Systems Engineering, the applicable quality standards for Systems Engineering, the location of the problem (translating *stakeholder needs* into *system requirements*) in Systems Engineering's context, and the boundaries of this thesis.

1.2 Systems Engineering

What is Systems Engineering?

Systems Engineering can be described as *"an interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life."* (ISO 24765, 2017, p. 457)

This *life cycle* of a system may be described as "*The evolution of a system, product, service, project or other human-made entity from conception through retirement*" (ISO 15288, 2015).

Different authors have defined Systems Engineering concept as follows:

- Faisandier (2014) states that "*Systems Engineering is the set of activities that allow the transformation of ideas (necessities or opportunities) into the description of an optimal system compared to all expectations and constraints*".
- SEBoK (2016) describes Systems Engineering *as "a pragmatic approach, inherently interdisciplinary, yet specialized*."

This holistic vision suggests studying the problem through a contextual analysis of the situation, exhaustively gathering the needs and later translating them into technical requirements, which will become the characteristics of a future system.

A little bit of history…

The term *Systems Engineering* was first used in the early 1940s in Bell Telephone Laboratories, New Jersey, United States of America, and its majors' applications were during World War II. In the late 1950s, SE was considered as a *science*, and was –and still is- taught in the Massachusetts Institute of Technology (MIT) (INCOSE, 2017).

In 1990, also in the United States of America, is founded the first professional association of Systems Engineering named NCOSE – National Council On Systems Engineering, and in 1995 it turned into INCOSE – International Council On Systems Engineering (INCOSE, 2017)

In France, Systems Engineering appears in the late 1980s, and in 1999 is created the AFIS - Association Française d'Ingénierie Système, INCOSE French chapter (Faisandier, 2014).

INCOSE and AFIS are associations without lucrative purposes, created with the objective of developing and spread the interdisciplinary principles and practices that allow the successful construction of systems (INCOSE, 2017; AFIS, 2017).

The INCOSE mission is defined as follows: "*To address complex societal and technical challenges by enabling, promoting, and advancing Systems Engineering and systems approaches*" (INCOSE, 2017).

Little by little, Systems Engineering has increased its presence in the world, and INCOSE increases the quantity and diversity of its members, who work collaboratively to advance Systems Engineering knowledge. Annually the INCOSE and the AFIS organize international congresses that allow knowledge sharing.

The point of view of both associations is very valuable for this research work due to its reliability level and innovation.

But, how is Systems Engineering related to *quality*?

International Quality Standards in Systems Engineering

As stated before, the general objective of this research work is focused on *the development of a methodology and its tools, based on Systems Engineering; in compliance with the quality standard ISO 15288 (2015), its aim is to ensure, as far as possible, that the stakeholder needs are understood, and are later correctly translated into system requirements, reducing risks of project abortion or delays, and adding value to the analysis and design process*.

Then, it is necessary to explain *why* specifically this standard, ISO 15288 (2015), is considered.

Quality

Over the years and more and more, companies have been forced to deliver high-quality products and services; therefore, *quality* has become a key element in the survival of enterprises and to maintain a competitive advantage (Topalovic, 2015).

Quality is defined as the "*ability of a product, service, system, component, or process to meet customer or user needs, expectations, or requirements* "(ISO 24765, 2017, p. 360).

As we can see, we may have *quality* products, *quality* services, *quality* systems, *quality* components, or *quality* processes. But when a new product, service, system, component, or process is designed, how can we be sure that it will meet the customer or user needs, expectations, or requirements? How can we assure its quality?

Trying to answer these questions, during the '80s appeared the concept *quality management* (Topalovic, 2015), described as the "*coordinated activities to direct and control an organization with regard to quality* "(ISO 24765, 2017, p. 362).

Quality assurance is the "*part of quality management focused on providing confidence that quality requirements will be fulfilled* "(ISO 24765, 2017, p. 360).

Then, *quality assurance* is a term that refers to "*all implemented pre-established and systematic activities within the framework of the quality system, to provide confidence that one entity will satisfy the requirements for the quality*" (ISO 24765, 2017).

In the actual competitive environment, delivering *quality* products and services is not enough for companies; it is also necessary to provide a *proof*, internationally recognized, that they have effectively implemented the *quality management system (QMS)* inside their organization.

A "*Quality Management System* can be defined as "*the organizational framework whose structure provides the policies, processes, procedures, and resources required to implement the quality management plan. The typical project quality management plan should be compatible with the organization's quality management system*" (ISO 24765, 2017, p. 363).

Here some definitions are presented to better explain QMS term:

Policy: "*set of rules related to a particular purpose*" (ISO 24765, 2017, p. 328).

Process: "*set of interrelated or interacting activities that transforms inputs into outputs*" (ISO 24765, 2017, p. 337).

Activity: "*set of cohesive tasks of a process*" (ISO 24765, 2017, p. 9; ISO 15288, 2015, p. 3).

Procedure: "*information item that presents an ordered series of steps to perform a process, activity, or task*" (ISO 24765, 2017, p. 336).

Quality management plan: "*a component of the project or program management plan that describes how an organization's quality policies will be implemented*" (ISO 24765, 2017, p. 363).

Quality Standards

If an enterprise wants to demonstrate that it provides high-quality products and services, fulfilling customer requirements –as far as possible- and certain regulations, it applies international quality *standards* (Topalovic, 2015). There are several *standards*; they are the result of international expert consensus.

According to the International Organization for Standardization (ISO), "*a standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes, and services are fit for their purpose*. Then, ISO *ensures that products and services are safe, reliable and of good quality* "(ISO 24765, 2017).

The *standards* propose a model to follow to set and operate a *QMS*. These standards can be applied to any organization, and some benefits of applying them are: more efficient use of resources, improved risk management, and increase customer satisfaction through quality services or products (ISO 24765, 2017).

According to SEBoK (2016), there is a multitude of standards across a number of standards development organizations (SDOs) that are related to Systems Engineering and systems domains. See Table 1.

Table 1 Summary of Systems Engineering Standards. (Adapted from SEBoK, 2016)

Document ID	Document Title	SDO
ISO 9001	Quality Management Systems – Requirements	ISO TC 176
ISO/IEC/IEEE 15288	Systems and Software Engineering - System Life Cycle Processes	ISO/IEC/IEEE
EIA 632	Engineering of a System	TechAmerica
ISO/IEC 26702 / IEEE 1220	Management of the Systems Engineering Process	ISO/IEC/IEEE
ISO/IEC/IEEE 29148	Systems and Software Engineering - Requirements Engineering	ISO/IEC/IEEE
ISO/IEC/IEEE 15289	Systems and Software Engineering - Content of Life-Cycle Information Products (Documentation)	ISO/IEC/IEEE
ISO/IEC/IEEE 15026-4	Systems and Software Engineering - System And Software Assurance – Part 4: Assurance in the Life Cycle	ISO/IEC/IEEE JTC 1

AFIS (AFIS, 2017) states that AFIS and INCOSE consider ISO 15288 as the reference for Systems Engineering practices; as a consequence, this research work considers ISO 15288 (2015) as a reference too for the methodology quality assurance purposes.

The standard ISO 9001 *Quality Management Systems – Requirements* will be taken into account when designing the *tools* that will provide methodology support.

The *tools* that will be proposed in this research work may be adapted to become part of a QMS:

a) the methodology procedures and forms (*documents*)
b) the information resulting from the application of the methodology (*records*)

NOTE: according to ISO 9001 (2015) *"documented information is broken into two types, documents, and records. A form is a kind of document. When the form is filled out it becomes a record"* (Retrieved 23/08/2018 from http://the9000store.com/articles/documented-information-required-by-iso-9001/). Here are presented some definitions that will help to better understand the previous note:

Document: *"uniquely identified unit of information for human use, such as a report, specification, manual or book, in printed or electronic form"* (ISO 24765, 2017, p. 143).

Record: *"set of related data items treated as a unit"* (ISO 24765, 2017, p. 363).

1.3 Problem context and research focus

In the framework of the standard ISO 9001 (2015) *Quality Management Systems – Requirements*, this research work is focused on the sub-sub-clause 8.2.2:

Clause 8 - Operation: *Plan and control processes needed to meet the requirements for products and services*.

Clause 8.2 – *Requirements for products and services*

Clause 8.2.2 – *Determining the requirements for products and services*

(Retrieved 23/08/2018 from http://the9000store.com/iso-9001-2015-requirements/iso-9001-2015-operational-requirements/)

According to ISO 15288 (2015), there are four different kinds of processes that may be performed through every system life cycle (Figure 2).

This research work is focused on the *Technical Processes* that transforms the stakeholder needs into a product.

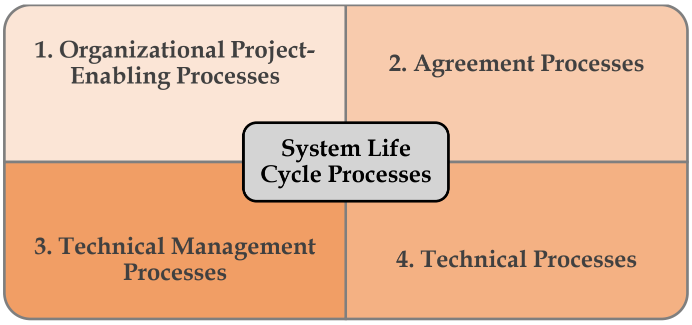


Figure 2 System Life Cycle Processes according to ISO 15288 (2015)

Technical Processes

Among the *Technical Processes*, this research work is bounded to the following two processes:

Stakeholder Needs and Requirements Definition Process (clause 6.4.2, ISO 15288, 2015): *"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. It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs. It analyzes and transforms these needs into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that is the reference against which each resulting operational capability is validated. The stakeholder requirements are defined considering the context of the system of interest with the interoperating systems and enabling systems."* (ISO 15288, 2015).

System Requirements Definition Process (clause 6.4.3, ISO 15288, 2015): *"The purpose of the System 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. This process creates a set of measurable system requirements that specify, from the supplier's perspective, what characteristics, attributes, and functional and performance requirements the system is to possess, in* *order to satisfy stakeholder requirements. As far as constraints permit, the requirements should not imply any specific implementation."* (ISO 15288, 2015).

Figure 3 shows that ISO 9001 and ISO 15288 are related with the Clause 8.2.2 and Clauses 6.4.2, 6.4.3 respectively, referred to system requirements definition.

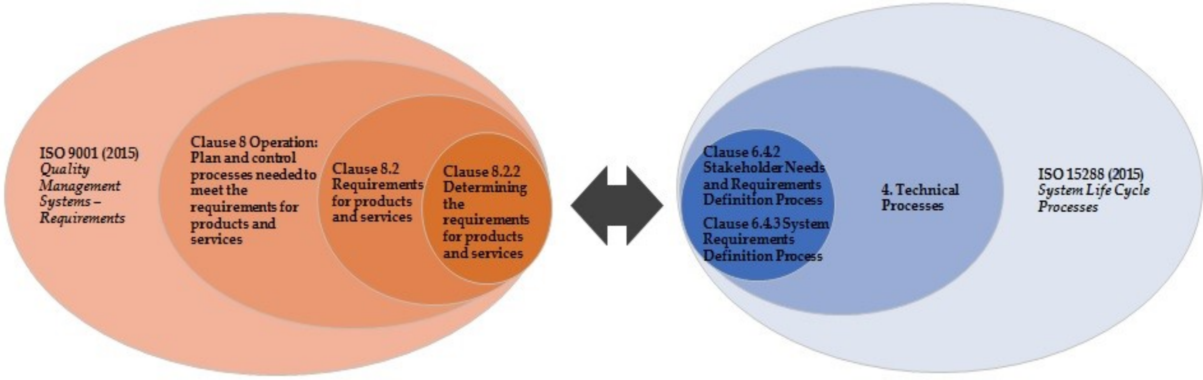


Figure 3 Requirements definition: ISO 9001 (2015) vs ISO 15288 (2015)

The two processes (*Stakeholder Needs and Requirements Definition* and *System Requirements Definition*) are conducted in the earliest stage of the system life cycle, the concept stage. Through time, different system life cycle models have been developed to facilitate the understanding of the life cycle of a system: how is conceptualized, developed, manufactured, used, maintained, and finally retired.

System Life Cycle Models

The generic system life cycle may be illustrated as shown in Figure 4.

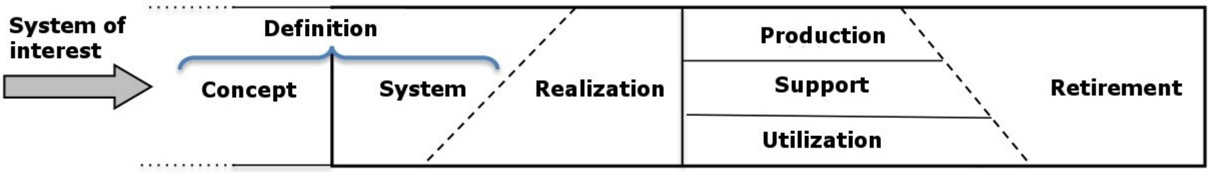


Figure 4 A Generic Life Cycle Model (SEBoK, 2016)

According to Forsberg & Mooz (1991), *"the technical aspect of the project cycle is envisioned as a "Vee", starting with User needs on the upper left and ending with a User-validated system on the upper right."* Ryan & Wheatcraft (2017) schematizes the Vee-Model as illustrated in Figure 5.

The two processes under study occur during the Concept stage of the system life cycle, the earliest stage, and they are directly linked to the cost of developing a new system. At the Concept stage, it is highly recommended that the set of requirements should be identified in order to avoid re-work, loses time, and additional costs in future stages.

"Design-it-now-fix-it-later" can be very expensive (Blanchard & Fabrycky, 2006), while more advanced is the progress of the project, the cost of design modification or alteration increases; it is calculated that 85% of the rework costs are originated by mistakes in the definition of system requirements (Kiritani and Ohashi, 2015; Szejka et al., 2014; De Weck & Willcox, 2010)

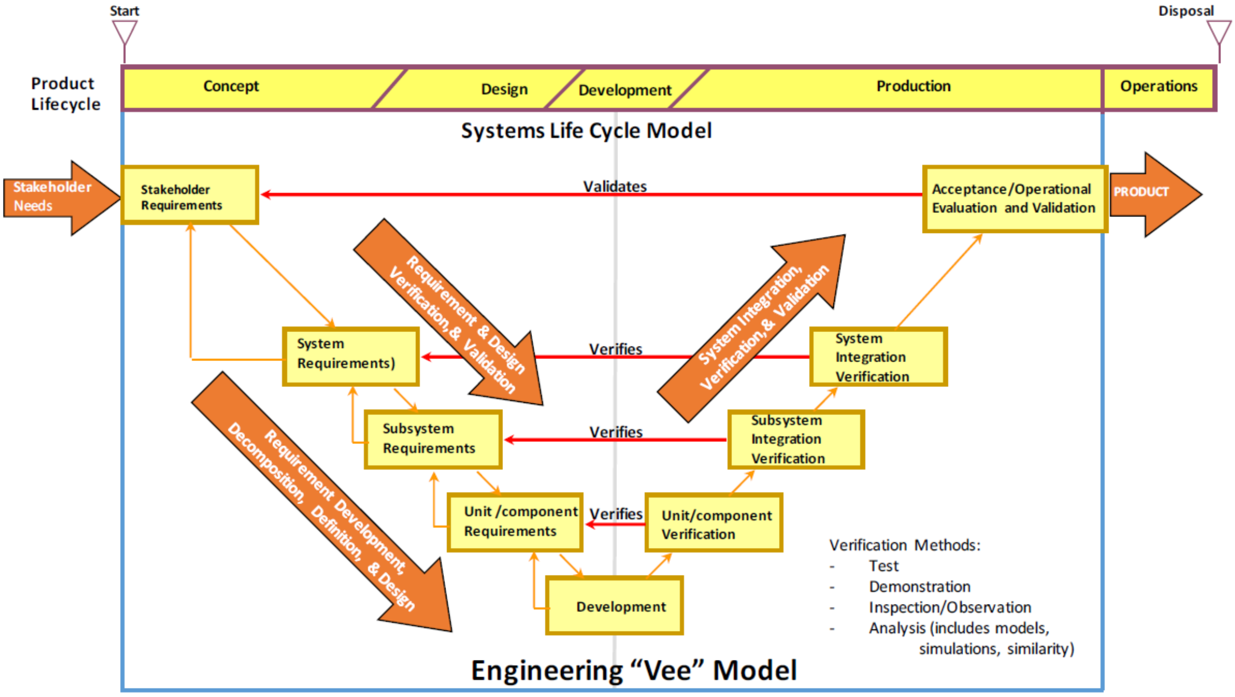


Figure 5 Engineering Vee-model (Ryan & Wheatcraft, 2017)

In Figure 6 it can be seen that at early stages of the system life cycle is less expensive and easier to change the requirements, compared to later stages, where the cost is higher as well the difficulty of change.

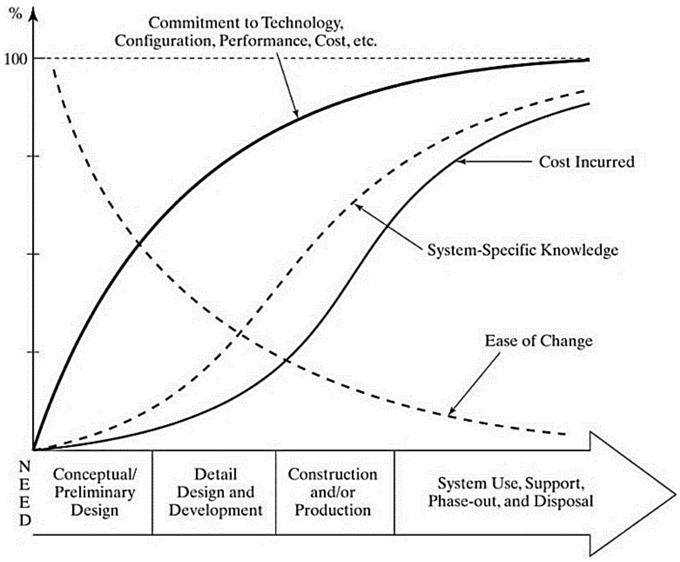


Figure 6 Life Cycle commitment, system-specific knowledge, and cost. (Blanchard & Fabrycky, 2006, p.46.)

This research is orientated to foresee the appearance of non-conformities during the activities of the definition of stakeholder requirements and their translation into system requirements, this way, it could be possible to increase the knowledge of the design earlier and retain the design freedom longer into the conceptual phase (De Weck & Willcox, 2010). See Figure 7.

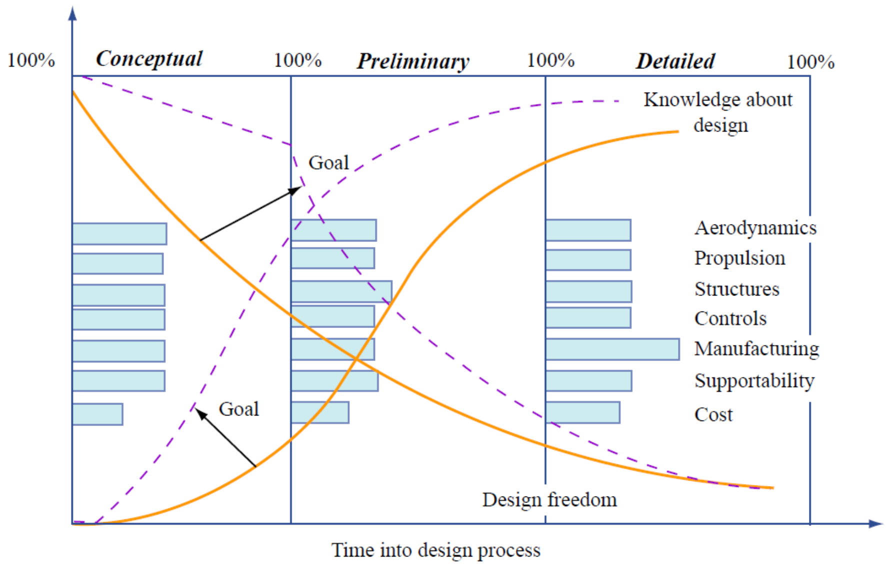


Figure 7 Design Freedom versus Knowledge. (De Weck & Willcox, 2010)

According to (Blanchard & Fabrycky, 2006), at the *Concept* stage, it is mandatory to ask the question words *what, who, when, why, where…* to assure that all the stakeholder needs are elicited and well interpreted, to later translate them into system requirements that will be transformed into a model or prototype that will meet all the stakeholder needs.

Then, the question word *what* should be used into a more specific level of questions that help for the identification of system requirements; for example: *what is expected from the system?* The same treatment is expected for the others question words like *which, how, when*, etc.; some examples are shown here: *Which functions must it do? Which are the primary and secondary functions? How can be eliminated the deficiencies found? When should it be done? Where should it be done? How frequently? etc.* (Blanchard & Fabrycky, 2006).

Each time a requirement is written, the following questions must be answered: *how will we verify that this requirement is reached? Which are the test and validation activities to be implemented?* (Blanchard & Fabrycky, 2006).

The discipline for studying requirements is Requirements Engineering.

Requirements Engineering

Requirements Engineering can be defined as an "*interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest*"; "*Requirements engineering is concerned with discovering, eliciting, developing, analyzing, determining verification methods, validating, communicating, documenting, and managing requirements.*" (ISO 24765, 2017, p. 381).

According to Faisandier (2012), Requirements Engineering can be considered as a *subsystem* of Systems Engineering; its activities include:

- Capturing stakeholder needs, goals, expectations, constraints and objectives.
- Preliminary identification of engineering system elements in terms of purpose, mission, expected services or operations, objectives, and physical interfaces with the context.

- The enrichment of engineering system elements through the analysis of preliminary engineering system elements generated.
- The translation of stakeholder needs into clear, concise and verifiable requirements.
- The potential refinement of the above-mentioned requirements in other more detailed.
- The analysis and translation of stakeholder requirements into system requirements
- The incorporation of system requirements into the selected alternative or solution.
- The management of stakeholder and system requirements through the life cycle, including their traceability.

From stakeholder needs to system requirements

As mentioned by Blanchard & Fabrycky (2006) and Faisandier (2012), the problem definition is the most difficult part of the process; it is essential to have a complete description of needs, expressed quantitatively and related with performance metrics whereas possible, and to state the correct and desired stakeholder requirements. This way the resultant system will meet the stakeholder needs.

The accurate identification of stakeholder needs is a necessity and represents a strategic, economic and commercial issue, the resources can be allocated at the adequate place and the focus of efforts will be correct. If the problem is not well defined, the consequences may be disastrous: at worst the solution could be rejected (Faisandier, 2012); in fact, one of the most important reasons of project failure is because the requirements were wrongly defined and consequently the resulting system is wrong too (Kiritani & Ohashi, 2015).

Then, stakeholder participation is essential when defining needs; when stakeholders collaborate with the analysis and design team, they help to clarify *what* should be solved, *what* they are looking for. The voice of the client should be heard, and the developers should answer in a consistent manner (Blanchard & Fabrycky, 2006). This way the system requirements could be defined in a right and concise form, avoiding ambiguities (Kiritani & Ohashi, 2015).

Authors like Acosta Flores (2002) suggest that the analysis and design team be interdisciplinary to have a wider and clearer vision. Blanchard & Fabrycky (2006) mention that the best way to do a satisfactory needs analysis is through a team conformed by the client, the end user (in case of a different from the client), the manufacturer and the principal suppliers.

The major difficulty inside an interdisciplinary team is the dialog among the specialists: they use different languages and models. These languages are differentiated by their degree of complexity; they are positioned in a relative classification that varies from *Natural* to *Formal* languages; one example is Systems Engineering language (SysML, Activity, Diagram) that is closer to *Natural* language in comparison with *Formal* language, see Figure 8.

Part of the answer resides in constituting a common language of engineering, or ontology, which standardizes the definitions and be accessible to every member of the team (Faisandier, 2014). To do this translation, the system engineer should know the semantic breaks propitiated by the different languages that the stakeholders and the designers speak.

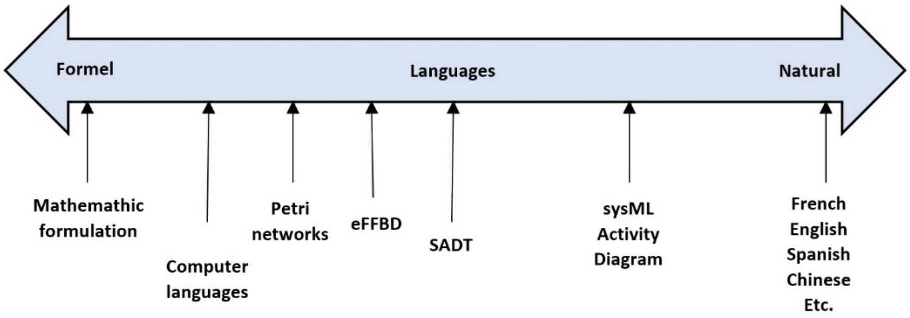


Figure 8 System Engineering language relative classification (Faisandier, 2014, p. 126)

Faisandier (2014) explains that during translating stakeholder needs to system requirements it can be observed that:

- Inside Systems Engineering community there are three kinds of engineers: the first group is focused in analyzing needs and requirements; the second group works in system architecture; and the third group integrates both spaces and has a holistic view of the system.
- Sometimes stakeholder needs are mixed with system requirements, and logical or functional architecture is not defined previously of building a physical solution, so, there is no model (functional, behavioral or temporal).
- Engineering activities are done discontinuously, the output of one activity does not represent the input of the next activity, there is not a predefined interaction among the activities and, consequently, the project development takes too much time; more important, the system may be rejected for lacking the requested requirements.
- The vocabulary or the terms used in Systems Engineering are semantically different from the activities to which one has related them historically.
- The activities are done in one order that appears illogical, it starts with constituents engineering before to study the logical and functional system architecture; the consequence is a system that does not meet the desired efficiency.

Figure 9 illustrates how Systems Engineering proposes to solve the problem from stakeholder needs to system requirements.

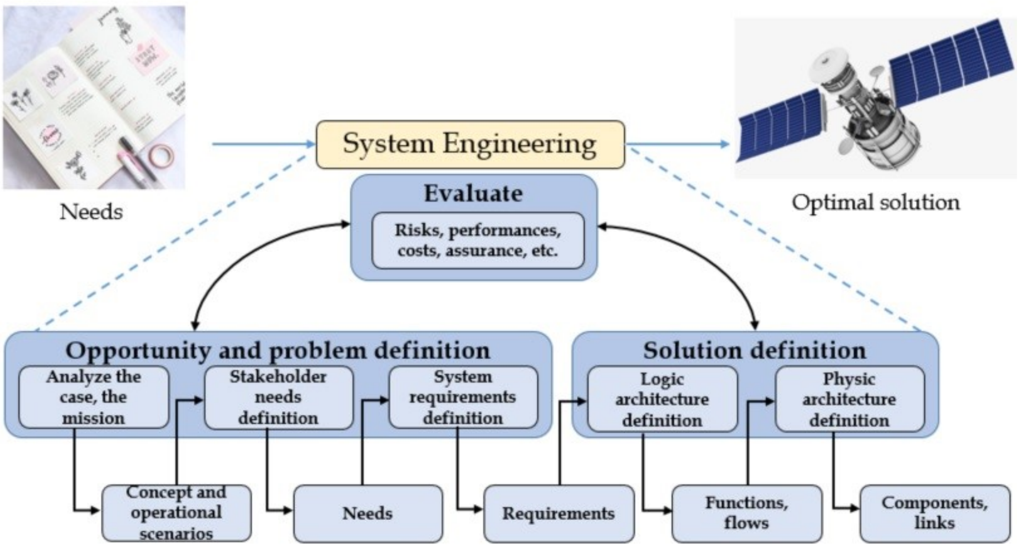


Figure 9 System definition activities. (Faisandier, 2014)

1.4 Conclusion

In this chapter, the context and problem formulation were introduced, starting with the definition of Systems Engineering, as well as the international and French associations focused on Systems Engineering continuing development and knowledge diffusion.

Systems Engineering technical processes were presented to better understand the limits of this research work, including different models to locate the problem. The Vee model was addressed as the preferred model inside the Systems Engineering community, and the importance of translating stakeholder needs into system requirements was highlighted.

The principal problems faced during stakeholder needs into system requirements were stated, and Requirements Engineering was also introduced.

International Quality Standard definition was presented, and ISO 15288 (2015) was stated as the quality standard reference for this research work.

In CHAPTER 2 Requirements Engineering the reader will find a deeper view of Requirements Engineering, starting with the definition of requirements, passing through their classification, quality characteristics, attributes, how to express them, and finally, the importance of requirement traceability through the system life cycle.

CHAPTER 2 Requirements Engineering

2.1 Introduction

As it has been stated in Chapter 1, Requirements Engineering (RE) can be defined as an "*interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest*" (ISO 24765, 2017, p. 381).

Nuseibeh & Easterbrook (2000), Cheng & Atlee (2007), and Femmer (2017) point five activities of Requirements Engineering: eliciting, analyzing, documenting, validating and managing requirements. Requirements are usually documented in the form of Requirements Engineering artifacts, such as free-text, uses cases or user stories; these artifacts are documented in natural language and they are an important factor for project success.

According to Badreau & Boulanger (2014) some recognized benefits of applying Requirements Engineering in Engineering Projects are:

- **Project success:** Requirements Engineering allows the early participation of interested parties, the completion of requirements considering only those that are feasible, as well as the management of their evolution throughout the development of the project.
- **Improvement of client need fulfillment:** Requirements Engineering allows increasing the number of satisfied needs because they are recognized in the early stages of the Project. On the left side of Figure 10 it can be seen that client needs are partially covered for the project; in contrast, on right side of the figure after applying Requirements Engineering, the area of satisfied needs is increased while unsatisfied client needs and useless system functions areas are reduced.

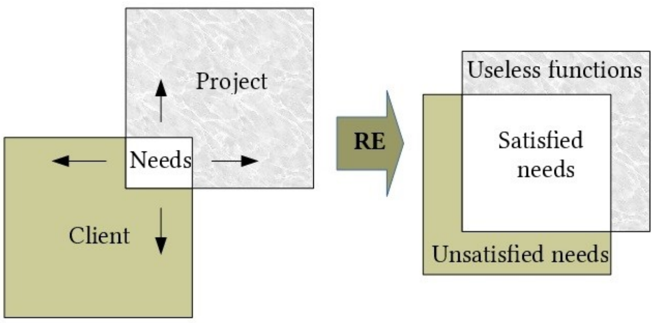


Figure 10 Maximize the need fulfillment (Badreau & Boulanger, 2014)

- **Higher client and user levels of satisfaction:** Requirements Engineering allows meeting the user and client needs, through taking needs into account at the time of conceptualization of the system. Figure 11 shows Kano's model, used to assess customer satisfaction; this model relates the customer satisfaction with the level of fulfillment of requirements. In this context, *time* is a key factor that may change the customer satisfaction level from very satisfied to dissatisfied, no matter if the system meets the requirements.

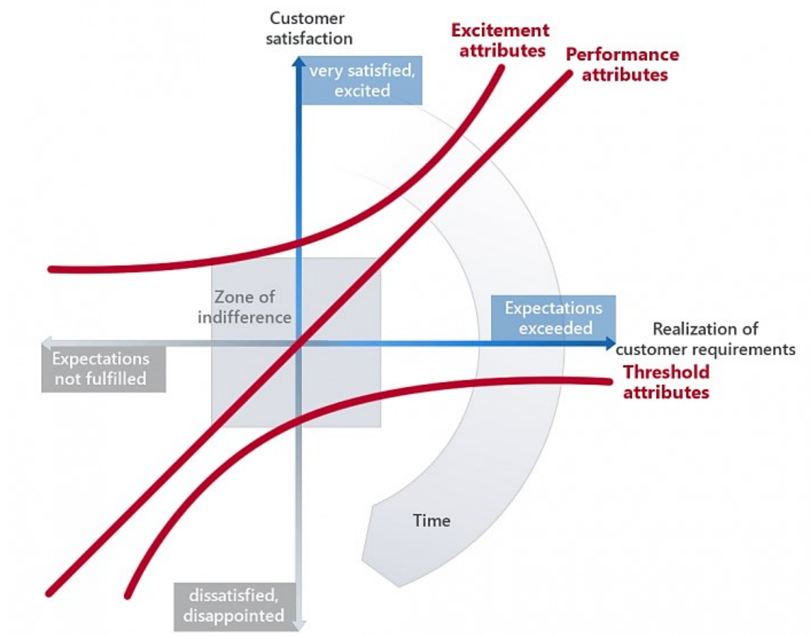


Figure 11 Satisfaction evaluation according to Kano (Badreau & Boulanger, 2014)

- **Early entry of the new product to the market:** Requirements Engineering helps to avoid delays in the introduction of new products, thanks to the early identification of stakeholder needs.
- **Several benefits to suppliers:** decrease of Project costs and Project risks, improvement of Project pilotage, product quality, and team communication and collaboration.

This research work will deal with *requirements*, which is why this chapter reviews the literature to broaden our understanding of them. It will be reviewed what is a requirement, how to express a *good* requirement statement, its attributes, and classifications. Additionally, this chapter surveys different RE process models, points out the importance of requirement traceability and its management through the system life cycle.

2.2 What is a Requirement?

Before it is explained the term *requirement*, it is important to share the term *stakeholder*. Femmer (2017) defines a stakeholder as "*a role, a person as a role, a person or an organization with a (direct or indirect) influence on a system*", including persons or organizations impacted by the system.

According to Ryan et al. (2015), there are several definitions of *need* and *requirement*. "*Needs are typically considered to be expectations stated in the language of those at the business management level or of stakeholders at the business operations level. Requirements are considered to be formal statements that are structured and can be verified and validated.*"(Ryan et al., 2015).

In spite of there is no agreement among authors or standards in the requirement definition, there are a number of common elements: requirements are product/system characteristics, conditions, and constraints that are unambiguous, testable and measurable.

For example, for Walden et al. (2015) the requirement definition is:

"A statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability."

Ryan et al. (2015) propose the following requirement definitions:

"An entity single thing to which a need or requirement refers: an enterprise, business unit, project, system, or system element (which could be a product, process, human, or organization)."

"A need is the result of a formal transformation of one or more concepts into an agreed-to expectation for an entity to perform some function or possess some quality (within specified constraints)."

"A requirement statement is the result of a formal transformation of one or more needs into an agreed-to obligation for an entity to perform some function or possess some quality (within specified constraints)."

The purpose of expressing a requirement is to communicate, as clearly as possible, the needs of an entity in formal language; this expression should be understood by those people who are going to implement the requirement in a system, those who will assess if the system meets the requirement, and finally those who will validate that the system meets the needs (Ryan et al., 2015).

Effectively communicating the needs is not so easy. The requirements communication model can be illustrated as in Figure 12. As it can be seen, there is a *need* in the mind of the *sender*, who tries to communicate it to the *receivers*. But here there is a problem to face: the *filters* cause that there is not a *single reality*; it is, each individual receive, interpret, and understand the information in different ways, so, *different realities* are constructed (Wheatcraft & Ryan, 2018).

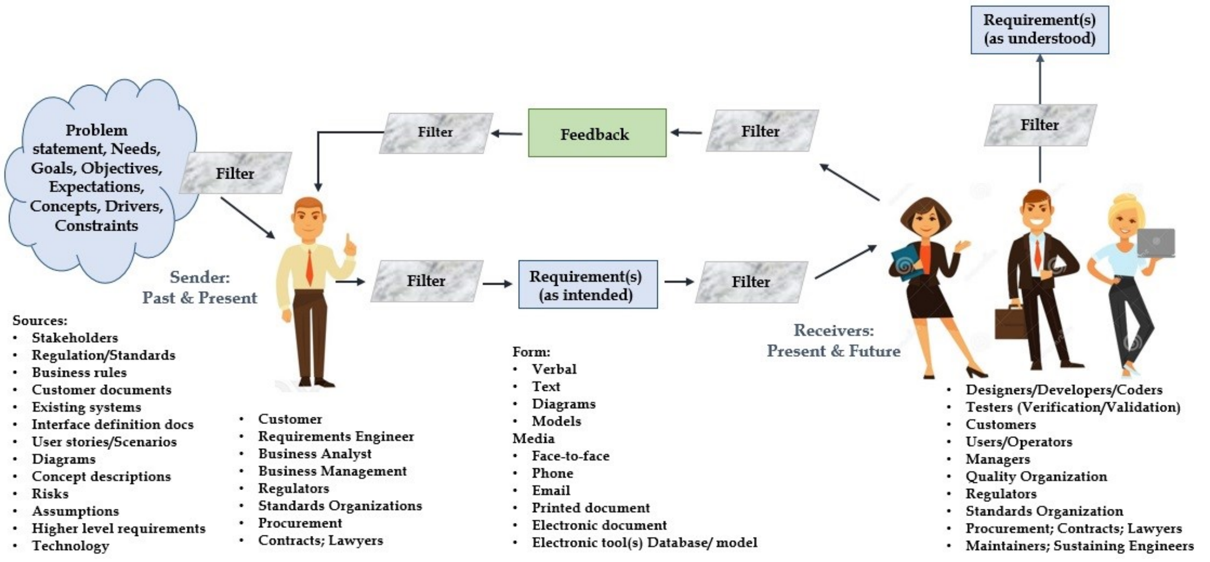


Figure 12 Requirements communication model (Wheatcraft & Ryan, 2018)

The consequence of this misinterpretation is that *senders* and *receivers* make wrong interpretations and conclusions concerning the initial *need*. The challenge in communicating requirements is that *requirements as intended* are the same that *requirements as understood* (Wheatcraft & Ryan, 2018).

The importance of the understanding of stakeholder requirements is because they are going to be translated into the requirements for the new system.

According to authors like SEBoK (2016), Ryan & Faulconbridge (2016), ISO 15288 (2015), Faisandier (2014), or Ryan (2013), requirements have different levels of abstraction:

- **Stakeholder requirements:** defined as *"a necessity or desire expected by an end user, formally drafted and expressed in terms of a client, of an end user; service, objective, capability expected from the future system by the end users."* (Faisandier, 2012).
- **System requirements:** defined as *"a statement that converts a stakeholder requirement into technical, usable terms for architectural and design activities using a semantic code (natural language, mathematical express, arithmetic, geometric, or software language, etc.)."* (Faisandier, 2012).

Then, it can be seen that a *stakeholder need* is first translated into a *stakeholder requirement*, and after the *stakeholder requirement* is translated into a *system requirement*. See Figure 13:

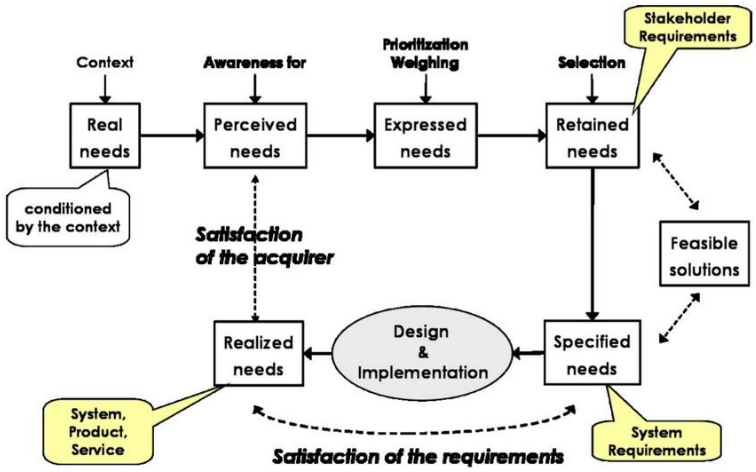


Figure 13 Cycle of needs (Faisandier, 2012, p. 82)

The following example was taken from Faisandier (2012, p. 155) and illustrates the different levels of abstraction: if the need is "*transfer a fluid*", the Stakeholder requirement will be "*System S transfers fluid from tank A to tank B* ", and the System requirements *(a) "System S extracts fluid from tank A" and (b)" System S fills up tank B*", which are functional requirements.

This translation helps the analysis and design team to conceptualize the needs into a design, later into an architectural model, and finally in a system that will satisfy the end-user needs.

Hull, Jackson, & Dick (2011) illustrate these translations as requirement *layers* in different domains, seen from different points of views, and playing different roles. Table 2 shows that the stakeholder requirements say what the stakeholder want, system requirements describe what the system will do to meet stakeholder needs, and finally, the architectural design states how the design will satisfy the system requirements.

Table 2 Problem and solution spaces (Hull, Jackson, & Dick, 2011)

Requirements layer	Domain	View	Role
Stakeholder requirements	Problem domain	Stakeholder's view	State *what* the stakeholders want to achieve through use of the system. Avoid reference to any particular solution.
System requirements	Solution domain	Analyst's view	State abstractly *what* the system will *do* to meet the stakeholder requirements. Avoid reference to any particular design.
Architectural design	Solution domain	Designer's view	State *how* the specific design will meet the system requirements.

2.3 How to express a Requirement?

In 1998, when the level of abstraction was not existent between stakeholder requirement and system requirement, Sommerville, Sawyer & Viller (1998) stated that "*The requirements …may be expressed in whatever notation is preferred by the sources. This means that there is no need for requirements sources to adapt their requirements to suit an inappropriate notation*."

Nowadays, we are used to find requirements expressed under the form of text, tables, diagrams, drawings, verbal, models, etc. (SEBoK, 2016). This way, requirements can be communicated face-to-face, by phone, e-mail, text-messages, printed or electronic documents, electronic tools, databases or models. Each organization decides what form and media is the one that best *fits* according to the idea or concept, the type of the system, domain, culture, people, and processes; nevertheless, the debate increases about what mean (form and media) of communication is the best (Wheatcraft & Ryan, 2018).

This research work is based on Systems Engineering, that is, the set of requirements should be generated and documented to easy their management, and these documents should be formally controlled (Wheatcraft & Ryan, 2018). Wheatcraft & Ryan (2018) express that well-formed, text-based requirement statements have these advantages:

a) Communication is improved due to text is universal
b) The power of expression of natural language
c) The wide accessibility to an electronic document format
d) Text-based requirements can contain attributes
e) Text-based requirements are understood easily in an agreement or contract
f) The verification formal process to system acceptance can be included in the agreement or contract

According to Badreau & Boulanger (2014), there are two principal rules to follow when understanding requirements and expressing as text: a) make short statements and paragraphs, and b) formulate only one requirement by statement.

For Ryan et al. (2015) *"a requirement expression comprises a requirement statement and a set of associated attributes."* For these authors, writing requirements "*is not simply an exercise in grammar, rather it is fundamentally an exercise in engineering*".

Mokammel et al. (2018) propose a pattern for writing requirements, it is the following:

1. System/system component (subject)
2. Necessity (modal verb)
3. Function/behavior/Characteristics (verb phrase)
4. Condition (adjunct)

For example (Mokammel et al., 2018): "*The air defense system* (1) *shall* (2) *be able to support joint operations* (3) *with long-range capabilities* (4)"

Shukla (2014) proposes to use of *negation sentences* when talking about a complex requirement, and write complementary requirements that restrict the possible deviated interpretation of the original requirement. For example, if the *need* is a comfortable chair, the requirement could be expressed as: a) *The user should feel relaxed even after prolonged usage* (affirmation sentence), or b) *The user should not feel any pain or cramp in his body, even after prolonged usage* (negation sentence); this way, the second possibility, the negation sentence, will easy the verification activity that should be carried out later (Shukla, 2014).

Requirements Engineering has a quality standard related, it is the ISO 29148 (2011) "Systems and Software Engineering - Requirements Engineering". This standard suggests that when writing textual requirements, the quality characteristics of *good* requirements should be considered. Moreover, the standard points that requirements should state *what* is needed, not *how*, this way requirements will not include design decisions.

De Weck, Rhodes, & Ross (2012) recognizes that "ilities" (a type of requirements related to quality) relate to changes in the system -function or form- over time. In this perspective, Ross, Rhodes, & Hastings (2008) propose four steps to formulate a requirement statement Figure 14:

1. Specify the location of the change agent (internal or external to the system)

2. Specify desired change effect (change or no change; level or set)

3. Perform system evaluation (calculate changeability metrics)

4. Specify subjective acceptability thresholds (conditions for "valuable")

The system shall be	______________	______________	in	______________	for less than	________.
	(change agent type) flexibly or adaptably	(change effects) scalable, modifiable		(system parameter) with range		(resources)

Figure 14 Example formulation of a requirement statement (De Weck, Rhodes, & Ross, 2012)

A lot of literature suggests how to write requirements, for instance, the Guide for writing requirements (2017) or Faisandier (2012). Similarly, it is possible to find indicated in literature the common mistakes when writing requirements and how to avoid them. For example, some mistakes are that requirements contain useless elements for understanding (pollution), the same requirement is repeated (redundancy), requirements are implicit, inaccurate, or omitted (lacks and weakness), requirements describe solution elements (solution-oriented), the same characteristic is defined in incompatible ways (contradiction), or requirements can be understood in several ways (ambiguity) (Faisandier, 2012). In Chapter 3 the reader will find more detailed the common mistakes when writing stakeholder requirements, and in Chapter 4 the common mistakes when writing system requirements.

ISO 29148 (2011) does not distinguish the level of abstraction between stakeholder and system requirements, the standard suggests:

- Avoid vague and general terms
- Avoid unbounded or ambiguous terms like:
    - Superlatives such as: the *best* or the *most*
    - Subjective language such as: *user-friendly, easy to use, cost-effective*
    - Vague pronouns such as: *it, this, that*
    - Ambiguous adverbs and adjectives such as: *almost, always, significant, minimal*
    - Open-ended, non-verifiable terms such as: *provide support, but not limited to, as a minimum*
    - Comparative phrases such as: *better than, higher quality*
    - Loopholes such as: *if possible, as appropriate, as applicable*
    - Incomplete references like: not specifying the reference with its date and version number; or not specifying only the applicable parts of the reference to restrict verification work
    - Negative statements (contrary to the idea of Shukla, 2014)

Badreau & Boulanger (2014) proposes the following recommendations to express requirements; in general: differentiate culture and vocabulary between the contractors and the project management to avoid incomprehension; these authors suggest to create a glossary for terms and definition references; and they suggest that this glossary should be centralized, accessible, compulsory, validated by stakeholders and defined from the start of the project with the responsibility for its maintenance.

2.4 Requirement Quality Characteristics

A requirement quality characteristic is a characteristic that describes how is written or expressed a requirement. There are quality characteristics for individual requirements and quality characteristics for a set of requirements.

Ryan et al. (2015) define *"a set of requirements is a structured set of agreed-to requirement expressions for the entity and its external interfaces documented in an Entity (Enterprise/Business Unit/System/System Element/Process) Requirements Specification (Document)."*

The quality characteristics of a requirement are really important because they will help the analysis and design team to verify that a requirement and a set of requirements are well expressed (Faisandier, 2012). This *expression* is not necessarily in a written form, there are requirements expressed as tables, drawings, pictures, etc. (SEBoK, 2016). The graphical language may include a lot of data, but it only can be used to express a few numbers of requirements; the disadvantage of using graphical language is the difficulty in managing requirement changes and traceability (Faisandier, 2012).

The quality standard ISO 29148 (2011) provides additional information on requirement characteristics, as individual requirement and as a set of requirements.

Quality Characteristics of an Individual Requirement

Several authors suggest which the quality characteristics of an individual requirement are; they sometimes have different points of view, as it appears in Table 3.

Table 3 Comparative of Individual Requirement Quality Characteristics by author

Individual Requirement Quality Characteristics	Definition: The requirement…	ISO 29148 (2011)	Faisandier (2012)	Badreau & Boulanger (2014)	SEBoK (2016)	Ryan & Faulconbridge (2016)	Guide for writing requs. (2017)
Appropriate to level	Must be appropriate to the level at which it is stated			•	•	•	•
Complete	Must not need further explanation	•	•	•	•	•	•
Conforming	Must conform to a standard formal structure				•	•	•
Verifiable	Can be verified that the system meets or possesses the requirement	•	•	•	•	•	•
Necessary	The system should not be able to function in the desired way without this requirement	•	•		•	•	•
Singular	Cannot be a combination of two or more requirements	•	•	•	•	•	•
Correct	An incorrect requirement will result in undesired system performance; it corresponds to a real need			•	•	•	•
Unambiguous	Only one interpretation of the requirement	•	•	•	•	•	•
Feasible	Must be achievable using existing technologies and manufacturing	•	•	•	•	•	•
Implementation free	The requirement states what is required, not how the requirement should be met	•	•	•			
Consistent (1)	Is free of conflicts with other requirements	•	•				
Consistent (2)	Is a consistent whole. The terms used in the requirement are the same and have the same sense			•			
Traceable	Is traceable to a specific source	•	•	•			
Concise	Written in the form of a single sentence that does not exceed some lines			•			
Clear	A reading of the requirement is enough for understanding it, the structure of the sentence is simple			•			
Accurate	All the elements used in the requirement are recognizable and completely characterized			•			
Up to date	The requirement reflects the common state of the system/its knowledge			•			

Coherent	The requirement is not in conflict (the requirement does not say a thing and its opposite) and it has coherence of the terminology (with the glossary)			•			

SEBoK (2016) highlights that traceability is no more considered as a quality characteristic of an individual requirement because traceability is not an intrinsic characteristic of a requirement, but it is more like a relationship between requirements.

As can be seen, these authors converge in some individual requirement quality characteristics as *complete, verifiable, singular, unambiguous*, and *feasible*, but differs in many others. Due to the impact of individual requirement quality characteristics in future activities like Verification, it is important to decide which among them are going to be considered.

In Table 4 are given the individual requirement quality characteristics, their definition, if they are considered or not in this research work, and the reason why.

Table 4 Considered individual requirement quality characteristics

Individual Requirement Quality Characteristics	Definition: The requirement…	Considered	Reason to consider the quality characteristic
Appropriate to level	Must be appropriate to the level at which it is stated	✓	For being consistent in the future with other design elements
Complete	Must not need further explanation	✓	It is understood very well with the elements that the requirement has
Conforming	Must conform to a standard formal structure	✓	In order to keep order and *neatness*
Verifiable	Can be verified that the system meets or possesses the requirement	✓	To realize if it was included or not in the design
Necessary	The system should not be able to function in the desired way without this requirement	✓	Because only required or needed requirements should be considered in the design
Singular	Cannot be a combination of two or more requirements	✓	Because only one requirement should be considered at a time, each one is important
Correct	An incorrect requirement will result in undesired system performance	✓	Because only *well understood (correct)* requirements will lead to the construction of the right system
Unambiguous	Only one interpretation of the requirement	✓	To avoid requirement misinterpretation
Feasible	Must be achievable using extant technologies and manufacturing	✓	If the requirement cannot be done, it is impossible to transform it into something real
Implementation free	The requirement states what is required, not how the requirement should be met	✓	It is considered in order to *open* the design alternatives, and because it is required by ISO 29148 (2011).
Consistent (1)	Is free of conflicts with other requirements	✗	It is considered in the *set of requirements* quality characteristics because *consistent* refers to relationships among requirements.
Consistent (2)	The requirement statement is a consistent whole. The terms used in the requirement are the same and have the same sense	✗	It is considered the same meaning that attributes *Complete* and *Unambiguous*
Traceable	Is traceable to a specific source	✗	It is not considered supported in SEBoK (2016): *traceability is no more considered as a quality characteristic of a requirement due to traceability is not an intrinsic characteristic of the*

			requirement, more like it is a relationship between requirements.
Concise	Written in the form of a single sentence that does not exceed some lines. Exactly shown, with precision	✗	It is not considered supported in SEBoK (2016): The requirement may be expressed as an individual statement, diagram, table, picture, etc. not only as a short statement
Clear	A reading of the requirement is enough for understanding it, the structure of the sentence is simple and does not use the literary subtleties The graphical language is not confusing	✗	It is considered the same meaning that *Unambiguous*
Accurate	All the elements used in the requirement are recognizable and completely characterized	✗	It is considered the same meaning that attributes *Complete* and *Unambiguous*
Up to date	The requirement reflects the common state of the system/its knowledge	✗	It is not considered because this characteristic is related to requirement management
Coherent	The requirement is not in conflict (the requirement does not say a thing and its opposite) and it has coherence of the terminology (with the glossary)	✗	It is considered the same meaning that attribute *Unambiguous*

Quality Characteristics of a Set of Requirements

In Table 5 are presented the different author point of views regarding quality characteristics of a set of requirements. The authors Ryan & Faulconbridge (2016) is not included in Table 5 because they do not express the quality characteristics for a set of requirements in the studied reference.

Table 5 Comparative of a Set of Requirements Quality Characteristics by author

Characteristics of a Set of Requirements	The set of requirements…	ISO 29148 (2011)	Faisandier (2012)	Badreau & Boulanger (2014)	SEBoK (2016)	Guide for writing requs. (2017)
Complete	The set of requirements needs no further amplification	•	•	•	•	•
Consistent	Does not have individual requirements contradictory	•	•	•	•	•
Affordable / Feasible	Can be satisfied by a solution that is feasible within life-cycle constraints	•	•		•	•
Bounded	Maintains the identified scope for the intended solution without increasing beyond what is needed	•	•			
Comprehensible	Must be written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part				•	•
Able to be validated	Must be able to be proven the requirement set will lead to the achievement of the entity needs within the constraints				•	•
Non-redundant	The same information or the same requirements shouldn't be expressed			•		

	several times among the set of requirements					

These authors coincide only in two of the quality characteristics of a set of requirements: *complete* and *consistent*. In contrast, they do not agree with many others. In Table 6 are given the quality characteristics of a set of requirements, their definition, if they are considered or not in this research work, and the reason why.

Table 6 Considered set of requirement quality characteristics

Quality Characteristics of a Set of Requirements	The set of requirements…	Considered	Reason to consider the quality characteristic
Complete	The set of requirements needs no further amplification	✓	It is considered because the set of requirements should show a wholeness
Consistent	Does not have individual requirements contradictory; Is the set of requirements free of conflicts among them?	✓	It is considered because all individual requirements should be coherent as a whole, without contradictions among them
Affordable / Feasible	Can be satisfied by a solution that is feasible within life-cycle constraints	✓	The set of requirements should be able to exist in reality
Bounded	Maintains the identified scope for the intended solution without increasing beyond what is needed	✓	The set of requirements should describe only the desired system, no more, no less
Comprehensible	Must be written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part	✓	The set of requirements should be understandable and clear
Able to be validated	Must be able to be proven the requirement set will lead to the achievement of the entity needs within the constraints	✓	The set of requirements should be able to be validated as a hole
Non-redundant	The same information or the same requirements shouldn't be expressed several times among the set of requirements	✓	The set of requirements should not repeat individual requirements

In addition to quality characteristics, requirements must also possess attributes.

2.5 Requirement Attributes

According to Ryan et al. (2015), *"an attribute is additional information included with a requirement statement, which is used to aid in the management of that requirement."*

Larson and Larson (2009), cited in Ryan et al. (2015), add that *"attributes provide information about the requirement, such as the source, the importance of the requirement, and other facts. Attributes aid in the ongoing management of the requirements throughout business analysis"* and highlight that *"the benefit of defining these attributes is to provide a means to validate the requirements as well as to provide metrics that will help to show the status of requirement development activities."*

In this research work requirement, attributes are considered as precious information because they will allow the analysis and design team to keep all valuable information, avoid loss of data and time, or rework gathering again the same information. They will help too in accomplishing requirement quality assurance and requirement managerial purposes.

Several Systems Engineering professionals like Faisandier (2012), Cuiller (2015) or professors and researchers like Ryan & Faulconbridge (2016) and Wheatcraft, Ryan, & Dick (2016) suggest some requirement attributes to their identification, management, and traceability.

ISO 29148 (2011) states as examples of requirement attributes: identification, stakeholder priority, dependency, risk, source, rationale, difficulty, and type.

Ryan et al. (2015), Guide for writing requirements (2017), and Wheatcraft & Ryan (2018) propose that attributes may be organized within four broad categories (see Table 7).

Table 7 Requirement attribute categories

a)	**Attributes to help define the requirement and its intent.** For example:	
	❖ rationale /reason	❖ benefit
	❖ verification method	❖ verification approach
	❖ validation method	❖ acceptance criteria
	❖ acceptance status	❖ parent requirement / need
	❖ source	❖ condition of use
	❖ states and modes	❖ business value
	❖ estimated cost	❖ actual cost
b)	**Attributes associated with the System of Interest (SoI) and system verification.** For example:	
	❖ SoI verification level	❖ SoI verification phase
	❖ SoI verification results	❖ SoI verification status
c)	**Attributes to help maintain the requirements.** For example:	
	❖ unique identifier	❖ unique name
	❖ short title	❖ originator/author
	❖ date requirement entered	❖ owner
	❖ stakeholders	❖ change board
	❖ change status	❖ version number
	❖ history	❖ approval date
	❖ target release	❖ related product release
	❖ date of last change	❖ stability
	❖ responsible person / assigned to a	❖ requirements verification status
	❖ requirement validation status	❖ status (of requirement)
	❖ dependants	❖ is-dependant on
	❖ conflicts	❖ trace to interface definition
	❖ trace to peer requirements	❖ interactions with external systems
	❖ physical characteristics	❖ related architecture elements
	❖ priority	❖ criticality
	❖ urgency	❖ difficulty
	❖ effort	❖ risk
	❖ key driving requirement	❖ questions associated with the req.
	❖ additional comments	❖ type/category
	❖ functional/performance	❖ supporting materials
	❖ model links	❖ environmental
	❖ facility	❖ ergonomic
	❖ compatibility with existing systems	❖ logistics
	❖ users	❖ training
	❖ installation	❖ location
	❖ person responsible for implementation	❖ status of implementation
	❖ transportation	❖ storage
	❖ the -ilities (quality)	❖ standards and regulations
d)	**Attributes to show applicability and allow reuse.** For example:	
	❖ region	❖ business unit
	❖ country; state/province	❖ application
	❖ market segment	

2.6 Requirement Classifications

Requirements have been classified through time and in different ways; but, which is the purpose of classifying requirements?

According to Palmer, Liang, & Wang (1990), requirements classifications/taxonomies are helpful as an approach to problem-solving and requirement analysis, to detect conflict among requirements, and that may be applied to complex tasks as verification and validation.

Sommerville, Sawyer & Viller (1998) propose an approach to *organize* system requirements that provides support for requirement elicitation; their approach identifies the concerns (goals) which affect the system, derive a set of questions to assure that the information required will satisfy the concerns (essential information and constraints), and finally elicit and negotiate the requirements which ensure that the system will satisfy the identified concerns.

Ryan et al. (2015), Faisandier (2014), SEBoK (2016), and Cuiller (2015) highlight that the importance of these requirement classifications is that they allow the consideration of all kinds of requirements, and ease the understanding of stakeholder requirements when translating into system requirements.

Authors like Glinz (2005), Badreau & Boulanger (2014), and Mabrok, Efatmaneshnik & Ryan (2015) classify product requirements in two big groups:

1) **Functional requirements (FRs):** they describe the expected services, what a system is supposed to or should do (Faisandier, 2012).
2) **Non-functional requirements (NFRs):** they are essential in the design process because they determine the technical specifications of the product that will be delivered. They impose constraints and capture the properties under the system must operate; for example its performance, cost, privacy, reliability, quality, etc. They describe the non-behavioral aspect of the system. (Mabrok, Efatmaneshnik, & Ryan 2015).

In Figure 15 is shown the requirement categories proposed by Badreau & Boulanger (2014).

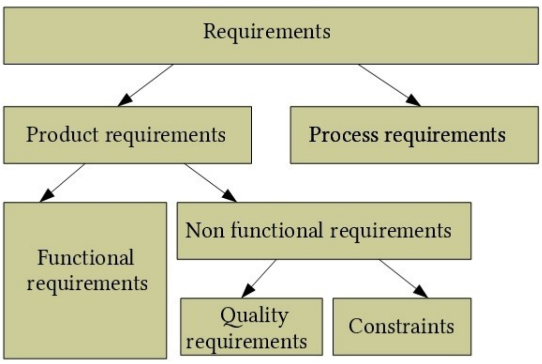


Figure 15 Requirement categories (Badreau & Boulanger, 2014)

In Figure 16 it is shown the classification of requirements, into functional requirements (FRs) and non-functional requirements (NFRs), proposed by Mabrok, Efatmaneshnik, & Ryan (2015).

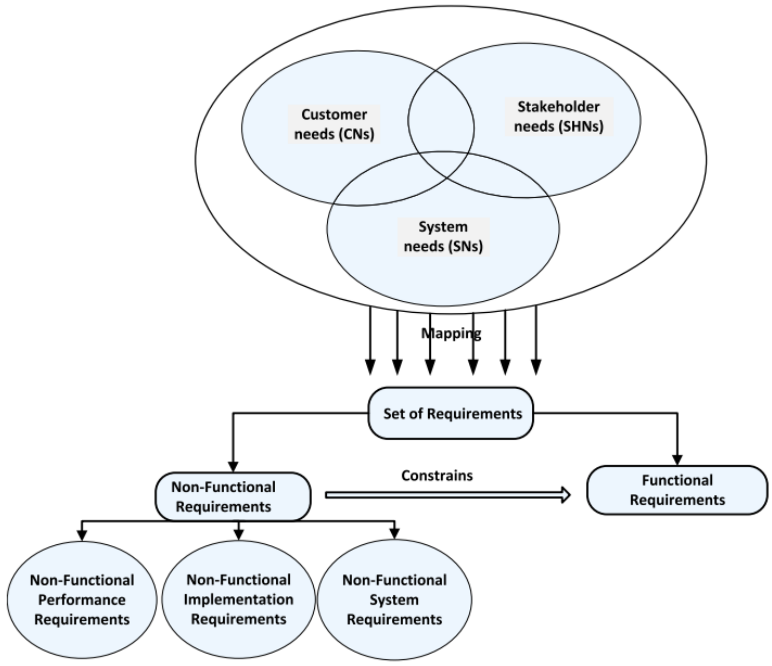


Figure 16 Classification of requirements in non-functional and functional requirements. (Mabrok, Efatmaneshnik, & Ryan, 2015)

Both functional and non-functional requirements are needed in the design process: the first ones to satisfy the need in functionality, the seconds to make possible the system and to satisfy the non-functional user expectations. "*Researchers have recently argued that there is no underlying theory as to why we ought to consider function as the most fundamental aspect of engineering design*" (Mabrok, Efatmaneshnik, & Ryan, 2015).

The inclusion of functional and non-functional requirements will allow the analysis and design team to make sure that the selected design satisfy (or not) both type of requirements. Figure 17 illustrates the axiomatic design theory (AD) for system design. It shows how the needs or customer attributes (CA) are placed at the *customer domain*, which will be mapped as functional requirements (FR), non-functional requirements (NFR), and constraints to characterize the future system (*requirements domain)*; the requirements, in turn, will be translated into design parameters (DP) (belonging to the *physical domain)*, which be transformed through processes characterized by process variables (PV) (*process domain*) to become a system that meets the customer expectations.

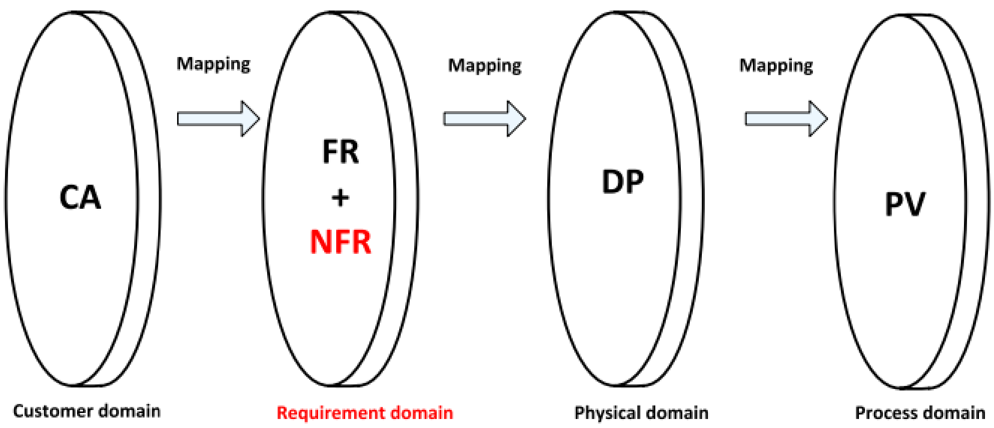


Figure 17 Four domains of the design world: Customer domain, requirement domain, physical domain, and process domain. (Mabrok, Efatmaneshnik, & Ryan, 2015)

There are several sub-classifications of non-functional requirements.

Glinz (2005) expresses that there is no consensus of "*what a non-functional requirement really is*", adding that there are divergent concepts for sub-classifying de NFRs. The author proposes a requirement classification based on four *facets* or concepts (see Figure 18): 1) kind (for example, if a requirement concerns to a function or its performance metric); 2) satisfaction (if the requirement is fully satisfied or only in certain degree at the time of verification); 3) representation (whether the requirement is represented operationally, qualitatively, qualitatively, or declaratively); and 4) role (the requirement may play three possible roles: prescriptive, normative, or assumptive).

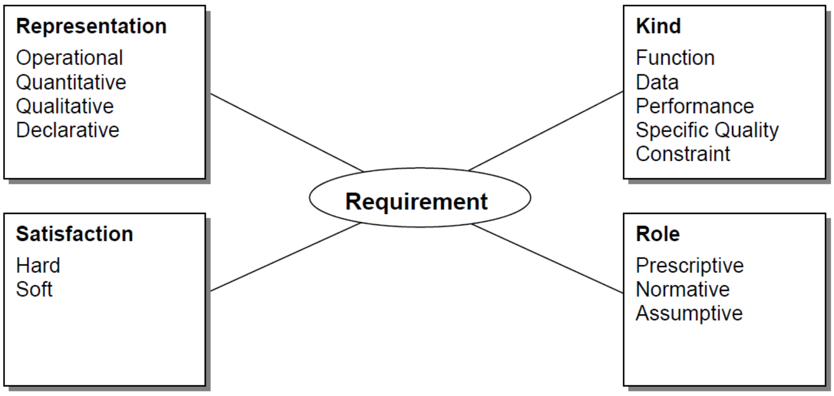


Figure 18 Faceted classification of requirements (Glinz, 2005)

Glinz (2005) expresses as benefits of his proposal: a) separating concerns clearly, b) identify the proper type of verification, and c) if the possible verification value is discrete or whether there is a range of acceptable behavior.

By their side, Ryan et al. (2015) summarize the lists of types of requirements as follows: *"business, stakeholder, user, project, product, environmental, unknowable, high-level, purposeful, functional (or behavioral or operational), non-functional (quality, security, modifiability, testability, reliability, portability, maintainability, availability, reliability, retention/purge, security, confidentiality and privacy, regulatory/legal/compliance, physical, safety, operational, scalability, data integrity, business continuity, usability, designability, efficiency, human engineering, human-factors, cultural and political, legal and others), performance, system, subsystem, component, verified, validated, and qualification, user interface, database, communications, external interfaces, performance, design constraints, quality, performance, interface, non-behavioral (portability, reliability, understandability, and modifiability), derived, goal level, domain level, product level, design level, primary, derived, role-based (customer, user, IT, system, security), engineering, transition, domain."*

De Weck et al (2012) talk about the "ilities": they are desired properties of the system that are not the primary system functional requirements but impact directly in those primarily functional requirements. The authors compiled a list of twenty "ilities" frequently encountered in their work on Engineering Systems. Figure 19 illustrates the frequency of "ilities" mentioned in journal articles and Google hits; as it can be seen, among the most popular "ilities" are quality, reliability, and safety.

The authors have observed that when the complexity of the system increases, as well increases the list of "ilities" for being considered. Moreover, they performed a study that revealed that exist some kind of *hierarchy* among the "ilities", some of them enable others. For example, changeability is supported by adaptability, flexibility, and modifiability; as long as modifiability is enabled by reconfigurability and modularity. See Figure 20.

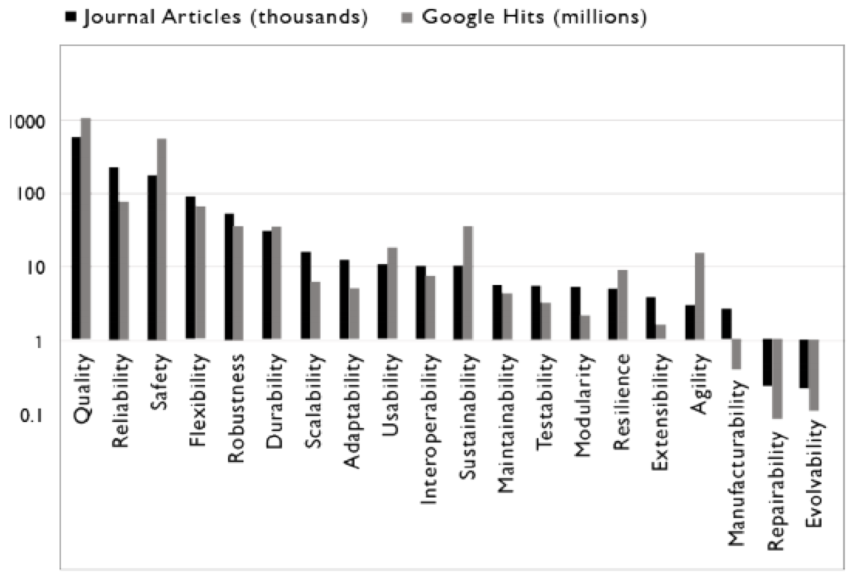


Figure 19 Frequency of ilities mentioned in journal articles and Google hits on the internet. (De Weck et al. 2012)

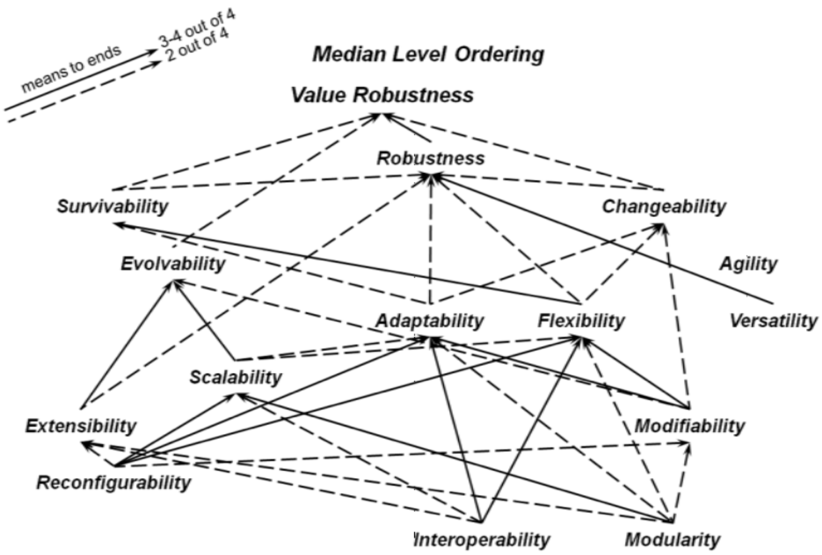


Figure 20 Aggregate median level ordering means-ends ilities hierarchy. (De Weck et al. 2012)

Grispos et al. (2017) mention that in the past few years researchers have proposed to integrate during system development what they call *forensic requirements*, concerning to detection, investigation, eradication, and recovery of incidents. These authors discovered that organizations appear to consider such requirements throughout the development lifecycle, but in many cases, they are considered after deployment resulting in increased development costs and delayed release times; that is why they recommend to include this kind of requirements during the *Concept* stage, analyzing the incident modes and incident scenarios.

In the article of Grispos et al. (2017) it is suggested that, in order to examine the causes of an incident, investigators need access to forensic data; then requirements should consider: availability of data for investigators, collection of data for investigators, secure storage of data for investigators, tamper-proofing forensic data, and examination of forensic data. However, the authors express that techniques for eliciting and analyzing these forensic requirements have not actually been proposed in the literature. See Figure 21.

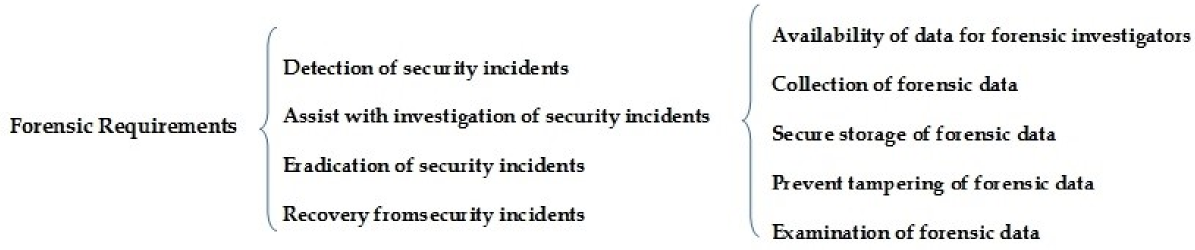


Figure 21 Forensic Requirement Classification proposed by Grispos et al. (2017)

There is another consideration to make: in his article "*Design for retirement*", Ryan (2014) talks about the "*end of life*" versus "*end of life cycle*" of a system. That is, the initial system -for some reason- is not needed anymore, and it may be sold to a second organization; now, the system enters into a second life cycle in which the new organization has to add to it new elements and to replace others. It could be possible that this *reuse* of the system occurs one or more times. See Figure 22.

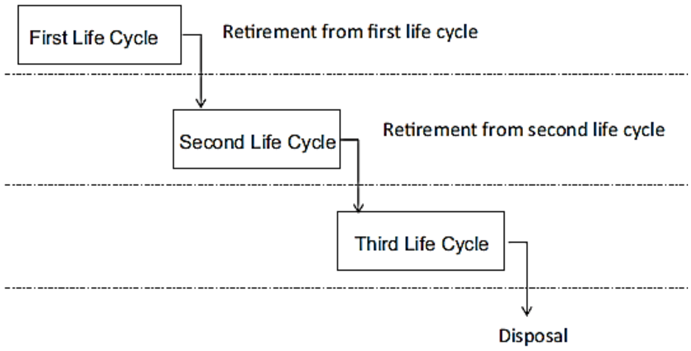


Figure 22 A system may be retired from a number of life cycles before final disposal at end of life (Ryan, 2014)

Ryan (2014) proposes that the system design should focus on the system retirement stage to include requirements related to the transition between the different life cycles of the system. In other words, it is necessary to consider during the system Concept stage: a) the reasons for system retirement, b) the potential retirement methods available (see Figure 23), and c) the design issues that will be present from the consideration of each retirement method.

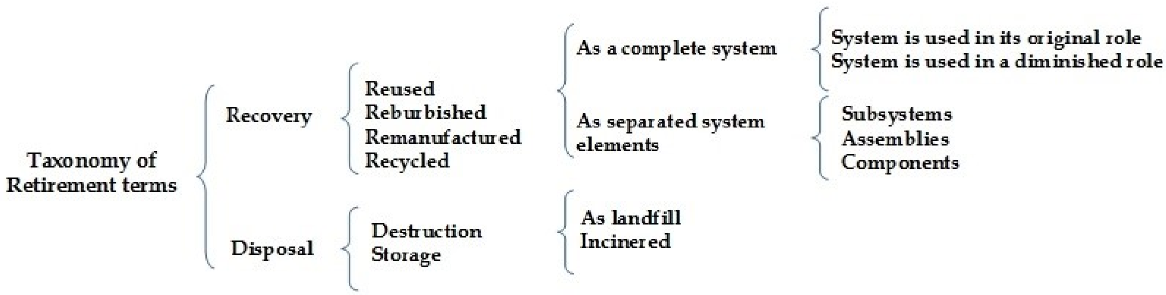


Figure 23 Taxonomy of Retirement terms proposed by Ryan (2014)

As stated before in Section 2.2 What is a Requirement? requirements have different levels of abstraction: stakeholder requirements and system requirements. In Chapter 3 and Chapter 4, the reader will find with more detail the state of the art corresponding to stakeholder and system requirement classifications.

Remembering that *"Requirements Engineering is concerned with discovering, eliciting, developing, analyzing, determining verification methods, validating, communicating, documenting, and managing requirements."* (ISO 24765, 2017, p. 381), through time, many authors have proposed different ways to model the Requirements Engineering process. In the next section, some of the most known contributions are shown.

2.7 Requirements Engineering Process Models

Batra & Bhatnagar (2017) conducted a comparative study of RE process models used in software engineering domain, with the objective of finding the vital aspects that contribute to RE process model selection. The compared model diagrams are shown here.

Figure 24 presents the Kotonya & Sommerville Linear Requirements Engineering Process Model, which includes repetition and overlapping among activities.

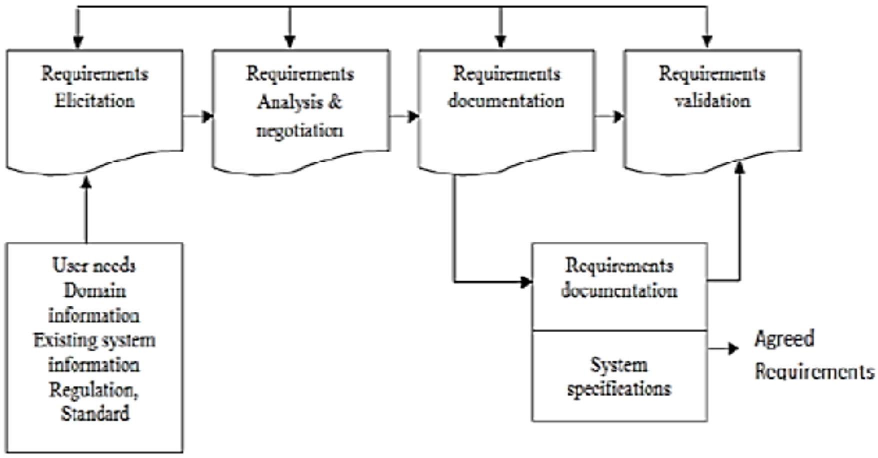


Figure 24 Kotonya & Sommerville linear requirements engineering process model (Batra & Bhatnagar, 2017)

Figure 25 illustrates the Macaulay Linear Requirements Engineering Process Model, which does not support overlapping of activities.

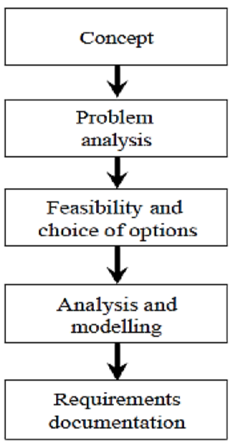


Figure 25 Macaulay Linear Requirements Engineering Process Model (Batra & Bhatnagar, 2017)

Figure 26 shows the Loucopoulos & Karakostas Iterative Requirements Engineering Process Model, which exhibits connections between the phases of Requirements Engineering.

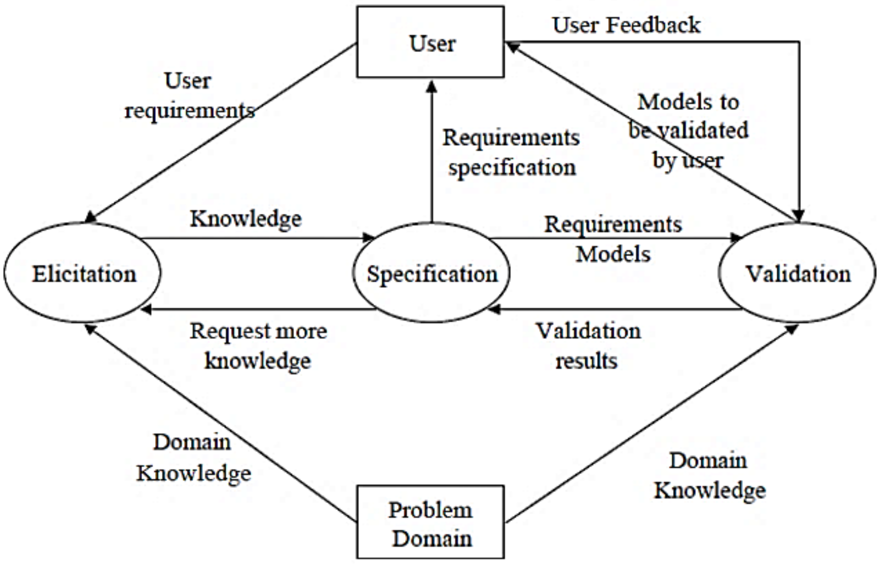


Figure 26 Loucopoulos & Karakostas Iterative Requirements Engineering Process Model (Batra & Bhatnagar, 2017)

Figure 27 represents the Kotonya & Sommerville Spiral Model of Requirements Engineering Process, which major objective is overcoming the consequences that affect the quality and cost of the project at different stages of software development.

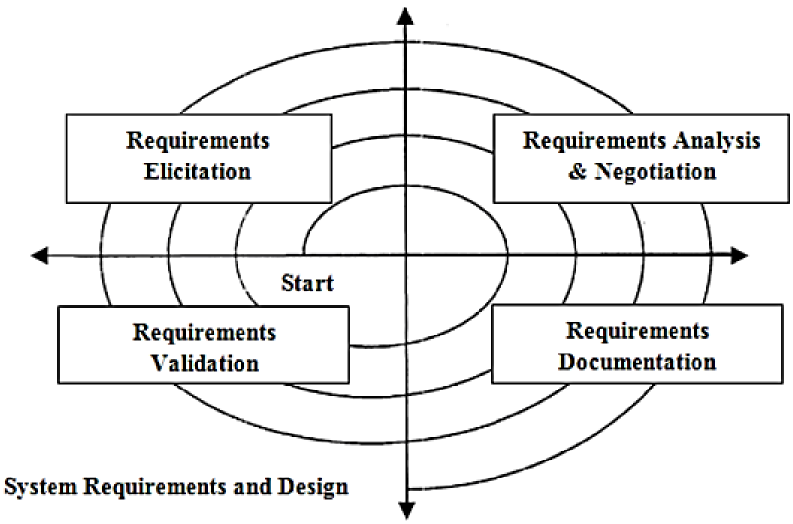


Figure 27 Kotonya & Sommerville Spiral Requirements Engineering Model (Batra & Bhatnagar, 2017)

Figure 28 shows the Ul-Arif, Khan, & Gahyyur Tools Cost-Benefit Analysis (TCBA) RE Process Model, which suggests using survey or interview methods depending on the number of stakeholders, includes the calculation of the return of investment (ROI) prior to the project and related costs, and risk management.

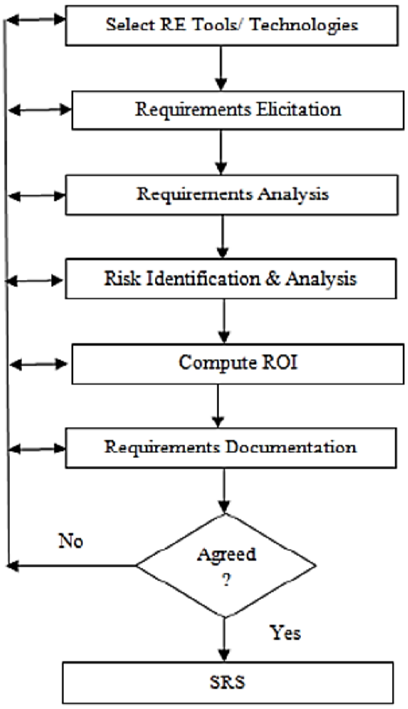


Figure 28 Ul-Arif, Khan, & Gahyyur Tools Cost-Benefit Analysis (TCBA) RE Process Model (Batra & Bhatnagar, 2017)

Figure 29 illustrates the Dhirendra Pandey & U. Suman Effective Requirements Engineering Process Model, which relates Requirements Engineering process to software development process, introduces a different kind of requirements to produce quality software products, and includes features for requirement management.

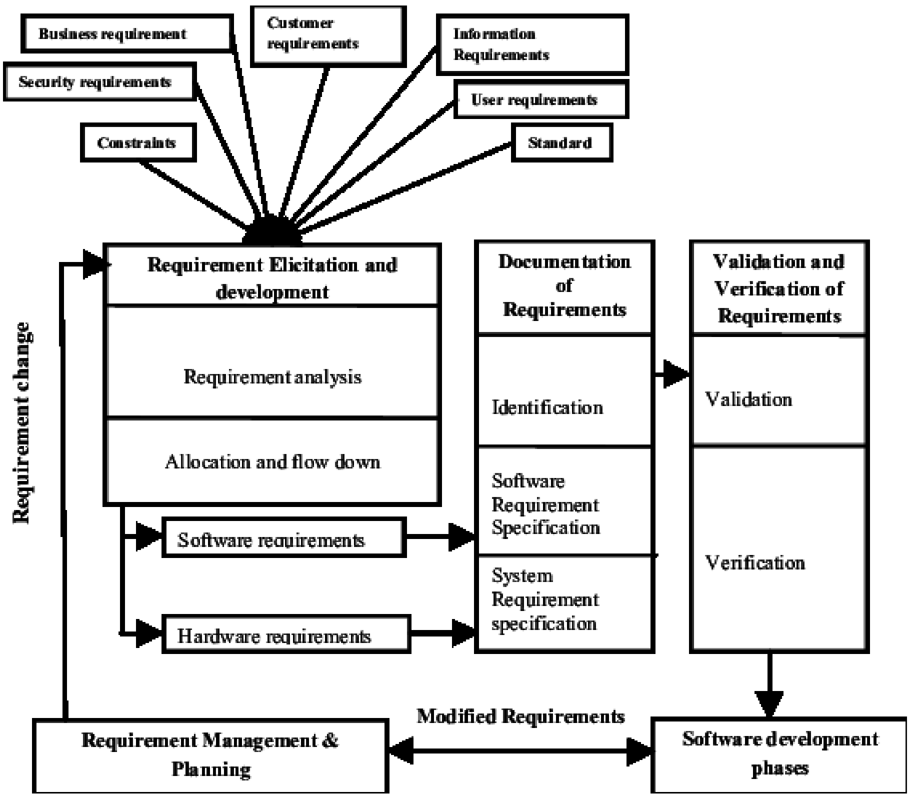


Figure 29 Dhirendra Pandey & U. Suman Effective RE Process Model (Batra & Bhatnagar, 2017)

Figure 30 presents the P.B.F. Arts Requirements Development & Management Model In Highly Turbulent Environments, which has three major phases: Intake phase, Startup phase (that suggests brainstorm technique for requirements elicitation), and Initiation phase (to prioritize requirements, obtain feedback, and validate them).

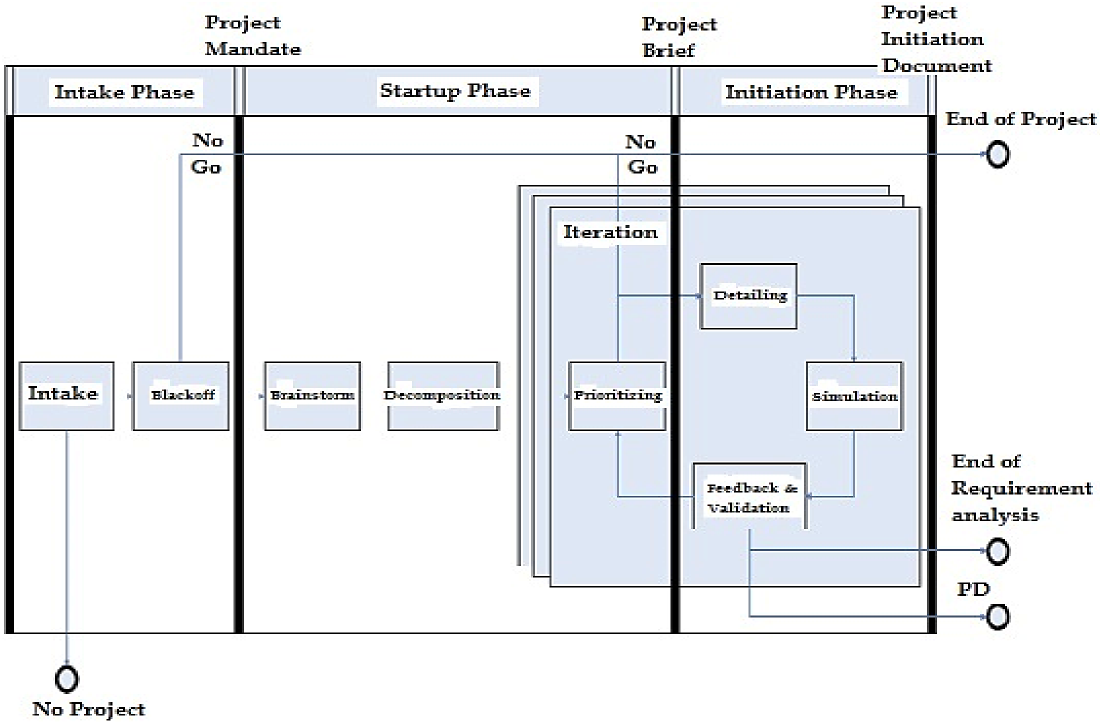


Figure 30 P.B.F. Arts Requirements Development & Management Model in Highly Turbulent Environments (Batra & Bhatnagar, 2017)

Additionally, it is considered the Bee Hive Model by K.S. Swarnalatha, G.N Srinivasan, & Pooja S Bhandary; this model accelerates the process of obtaining requirements to design the prototype; it consists of the following phases: background research, elicitation and analysis, prototyping, verification and validation, and requirement specification.

Batra & Bhatnagar (2017) compared these different methods considering aspects as linearity, support for changing requirements, iterative in nature, support for reverse engineering, risk assessment, criteria for application specific requirements elicitation techniques, requirements pre-processing, requirement prioritization, and effort estimation. Their findings can be seen in Table 8.

Table 8 Comparison of Requirements Engineering Process Methods (Batra & Bhatnagar, 2017)

Model characteristics	Kotonya & Sommerville linear model	Macaulay model	Loucopoulos & Karakostas model	Kotonya & Sommerville spiral model	Ul-Arif, Khan & Gahyyur model	Pandey & U. Suman model	P.B.F. Arts model	Swarnalatha, Srinivasan, & Bhandary model
Linearity	✓	✓	✗	✗	✗	✗	✗	✗
Support for changing requirements	✗	✗	✗	✓	✓	✓	✓	✓
Interactive in nature	✗	✗	✓	✓	✓	✓	✓	✓
User feedback	✗	✗	✓	✓	✗	✓	✓	✗
Support for reverse engineering	✗	✗	✗	✓	✗	✗	✗	✗

Risk assessment	✗	✗	✗	✓	✓	✗	✗	✗
Criteria for application specific requirements elicitation technique	✗	✗	✗	✗	✗	✗	✗	✗
Requirements pre-processing	✗	✗	✗	✗	✗	✗	✗	✗
Requirements prioritization	✗	✗	✗	✗	✗	✗	✓	✗
Effort estimation	✗	✗	✗	✗	✗	✗	✗	✗

As it can be interpreted from Table 8, and supported with the conclusion of their study: *"Researchers have made considerable advancement in the area of requirements engineering but still development is needed…further work is required such as requirements preprocessing, risk management, requirements prioritization, application of a specific elicitation technique, etc."* (Batra & Bhatnagar, 2017). According to this information, the Kotonya & Sommerville Spiral Requirements Engineering Model (see Figure 27) seems to be the best among them; nevertheless, they fulfill only five of ten desired characteristics.

In Quality domain, there is another framework to translate stakeholder needs into system requirements, it is the Quality Function Deployment method (QFD).

According to the QFD Institute (2017) "*QFD is a methodology for customer needs analysis and solution development. QFD continues to evolve, from its early days as a technique for assuring quality before manufacturing.*"

Mazur (2012) explains that the QFD method was developed in Japan in the 1960s to assure that not only *negative quality* is prevented, but *positive quality* is enhanced. This approach recommends:

- Assuring quality is a team approach
- Customer-driven quality required acquiring and analyzing the voice of the customer, in order to determine what matters most
- Different customer segments have different needs with different strengths

The heart of the process is the QFD matrix, known as the *house of quality* (HoQ) developed by Dr. Yoji Akao in 1972 (Kossiakoff et al., 2011). In the HoQ it can be seen the *customer needs* (stakeholder requirements), their *priorities* or ranking, the *product characteristics* (system requirements), the *co-relation* between different product characteristics, the *relationship* between customer needs and product characteristics, additional *customer analyses,* and *overall weighting including technical analysis*. See Figure 31.

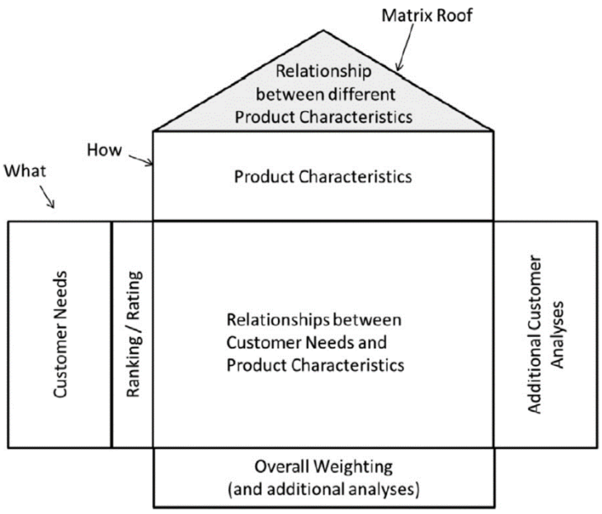


Figure 31 Generic House of Quality (Schillo, Isabelle, & Shakiba, 2017)

During 1980s QFD was applied in the U.S. automotive parts industry, and in the 1990s efforts were focused on creating elaborated matrix or *houses of quality* to help the multi-functional teams visualize complex cause-effect relationships among users, developers, builders, and deliverers of products and services. See Figure 32.

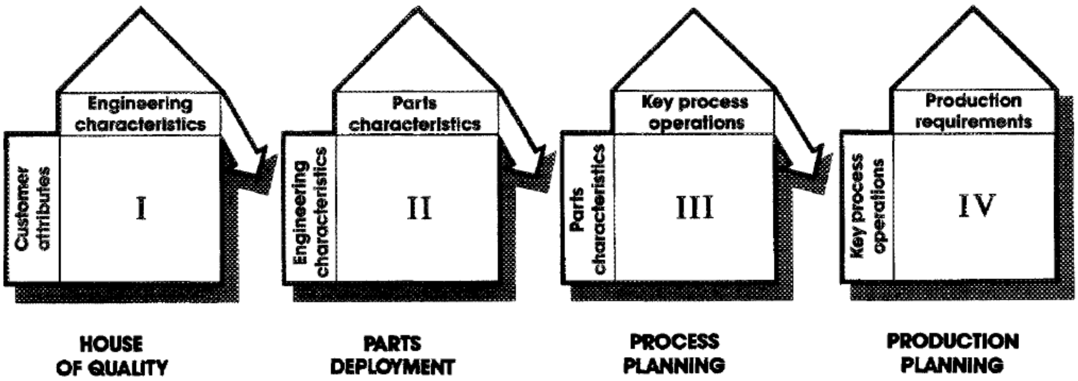


Figure 32 Linked houses of quality (Hauser & Clausing, 1988)

The application of these matrices helped significantly to improve the effectiveness of aligning product designs to the customer requirements; nevertheless, the complicated matrix reduced the effectiveness in practice, leading to analysis and design teams demotivation, and in many cases, the method was abandoned (Mazur, 2012). See Table 9.

Table 9 Advantages and disadvantages of Classical QFD (Mazur, 2012)

Advantages	Disadvantages
Exhaustive coverage	Exhausting to complete
Timeless	Time-consuming, risk of giving up
Easily reusable	Not useful until completed, but only a small part is actionable. Not a good use of scarce resources
Prevents ítems "falling through the cracks"	Predetermined elements lock in the existing paradigm
	Problematic math

With the influence of the *Lean* philosophy, a Blitz QFD® approach emerges which ensures that maximum value can be obtained with a minimum of resources, Blitz QFD is the lean equivalent of QFD (Bylund, Wolf, & Mazur, 2009). This approach developed by Richard

Zultner offers four significant improvements compared to the traditional QFD method (Mazur 2012):

a) Efficiency & speed of analysis can get improvements immediately, no need to wait until the analysis is completed
b) Establishing true customer needs and values, a sense of focus on high-value items
c) Relative/Proportional values of the priority
d) Lean design – effort focused on critical customer business goals and requirements
e) Faster time to complete and math issues addressed

Schillo, Isabelle, & Shakiba (2017) state that "*QFD provides a systematic consideration of all relevant stakeholders, a rigorous analysis of the needs of stakeholders, and a prioritization of design features based on stakeholders needs*".

Mazur (2002) indicate that the Blitz QFD consists of 7 steps illustrated in Figure 33:

1. Project Definition (get the voice of the customer)
2. Statement Analysis (classify verbalizations)
3. Situation Analysis (structuring customer needs)
4. Goal Analysis (analyze the client needs structure)
5. Prioritize customer needs
6. Project Strategy (deploy priority needs)
7. Downstream Deployments (7 MP + Tools) (analyze priority relationships in detail only)

Mazur (2012) and Bylund, Wolf, & Mazur (2009) agree that the QFD method is a generic process that is suggested to be adapted (tailored) to the specific objectives of each organization. It begins with the definition of the project and ends with the Design & Concept Selection.

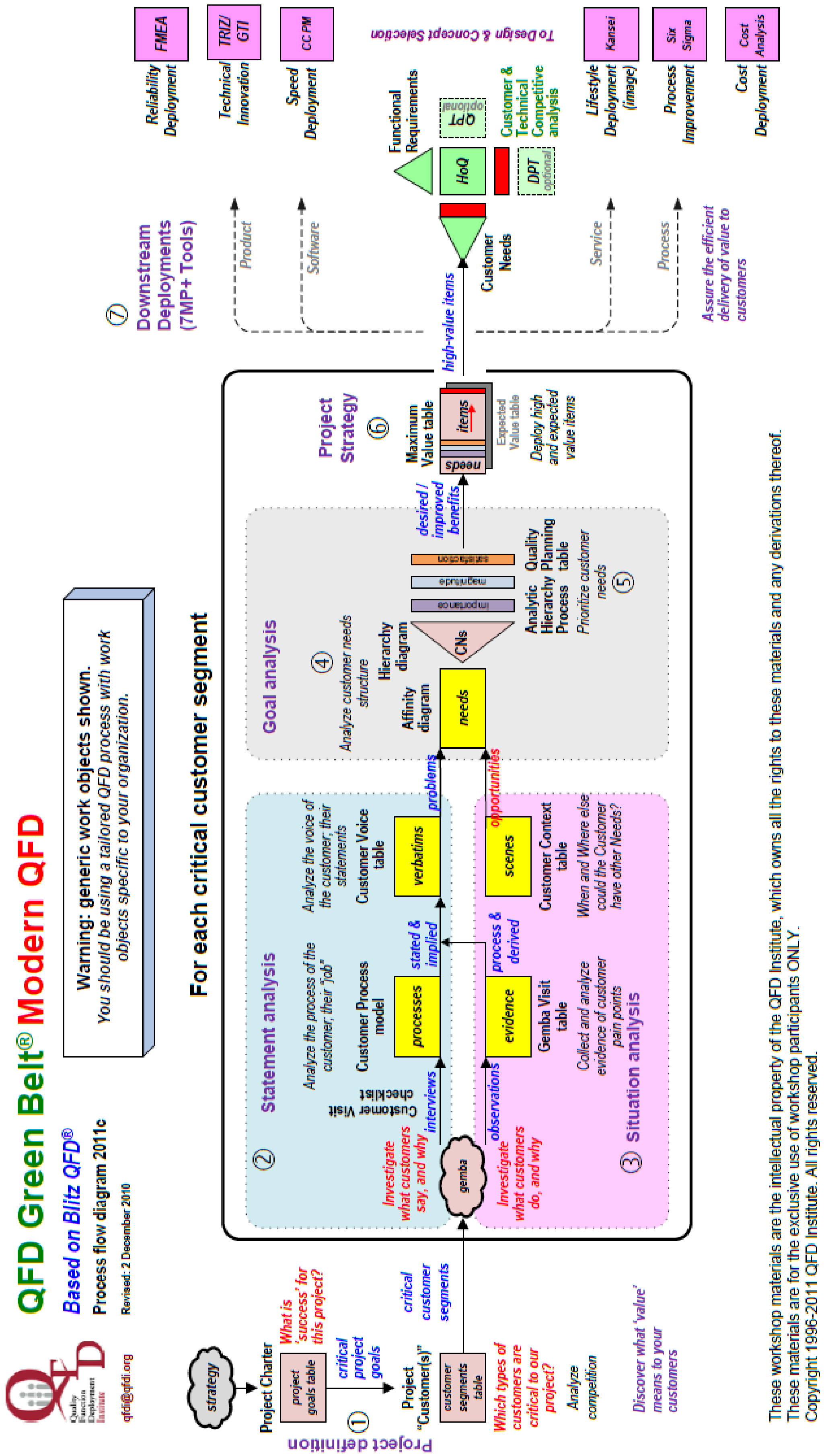


Figure 33 Generic Blitz QFD process flow diagram (Mazur, 2012)

It is important to highlight that the HoQ has had an important improvement: the co-relationship between different *customer needs* on the left side (Mazur, 2012). See Figure 34.

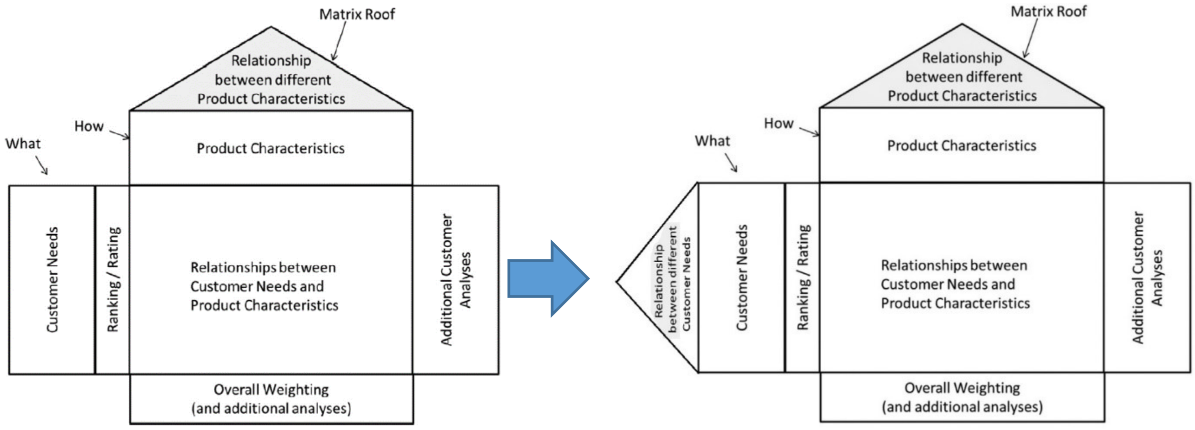


Figure 34 Generic QFD HoQ matrix (adapted from Schillo, Isabelle, & Shakiba, 2017)

In Systems Engineering domain, Ryan (2013) proposes a generic model to translate a company's strategic intent into the definition of a system. See Figure 35.

The model is divided into two main sections: *Needs View* and *Requirements View*, which in turn are subdivided into five levels or layers (layers) that are associated with generic organizational views: Enterprise, Business Operational, System, and subsystem.

The initial entry is defined by the Concept of Operations (ConOps) that defines the enterprise strategies.

In each layer, the needs of each organizational view are translated by analysis into requirements. *Enterprise strategies* are translated into *business needs*; *business needs* are translated into *business requirements* (BRS) and *stakeholder needs*; *stakeholder needs* are translated into *stakeholder requirements* (StRS); *stakeholder requirements* are translated into *system requirements* (SyRS); and finally, *system requirements* are translated into *subsystem requirements.*

With this model, the requirements can be traced back to its previous layer.

The expression of the requirements differs from layer to layer and there are rules for expressing them; as more progress is made in the model the level of abstraction increases and the resulting requirements become more and more specific.

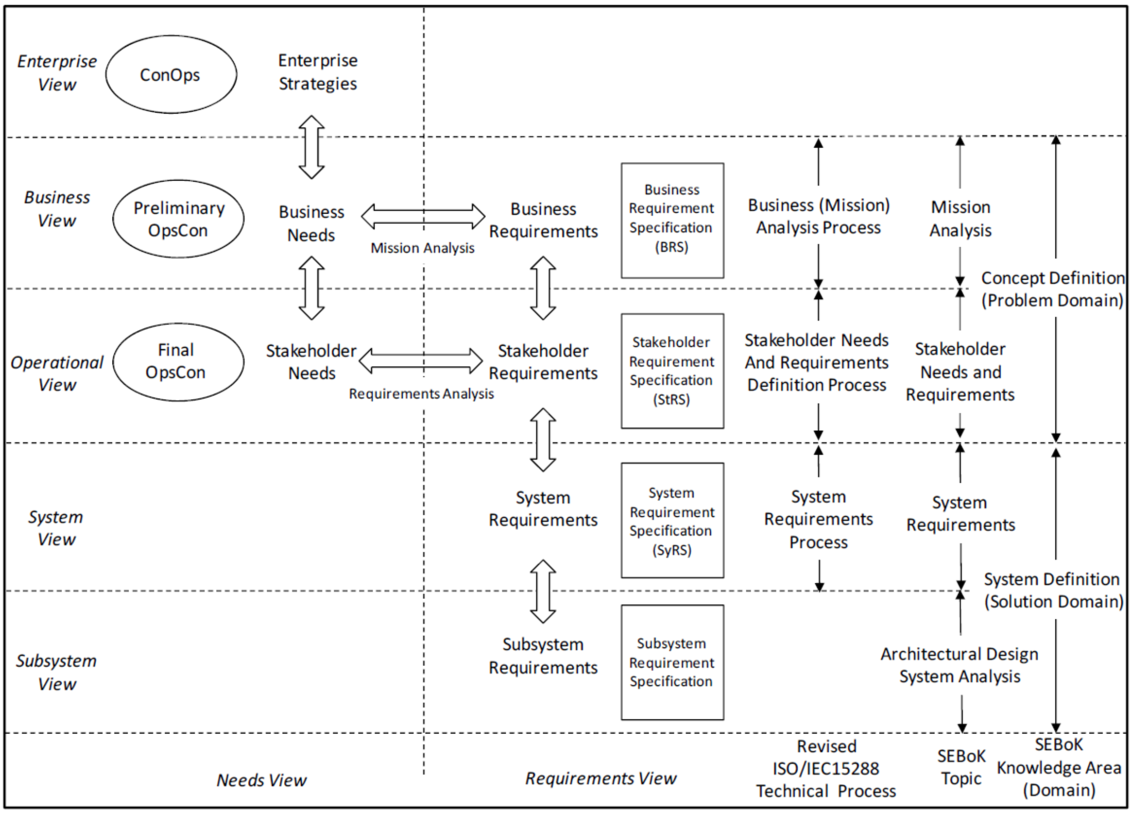


Figure 35 Views, domains, processes, and artifacts associated with needs and requirements (Ryan, 2013)

2.8 Requirement Traceability

Once the requirements are expressed, it is suggested by several authors to keep traceability.

Faisandier (2012) defines: "*traceability of requirements consist of tracing information from the origin of the Stakeholder Requirements of the top level to the lowest level of the System of Interest hierarchy and vice-versa*".

Ryan & Faulconbridge (2016) mention that traceability ensures knowledge of where the requirement comes from, what requirements it is related to and what requirements are derived from it. Faisandier (2012) explains that traceability is also used to analyze the impact on the system of interest (SoI) when a change in a requirement is required.

In our case study, traceability is important as it helps to visualize that each stakeholder requirement is translated into at least one system requirement. According to Faisandier (2012), the traceability of requirements is ensured through the different system blocks, respecting the repetitive sequence for each level of decomposition. See the following Figure 36.

By its side, SEBoK (2016) recommends establishing *bi-directional traceability between requirements at adjacent levels of the system hierarchy*; nevertheless, SEBoK (2016) does not give a definition or meaning of this term, neither provides an explanation.

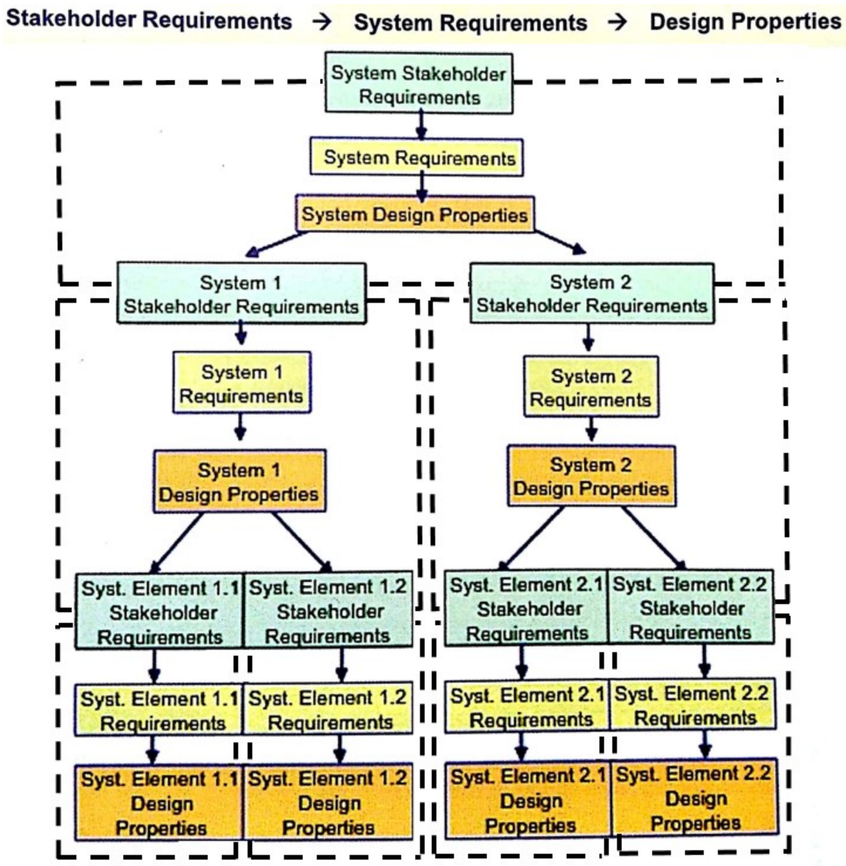


Figure 36 Requirement traceability between system-blocks (Faisandier, 2012, p.120)

2.9 Conclusion

In this chapter Requirements Engineering has been presented, including the definitions of *need* and *requirement*. Also, relevant information has been written regarding how to express a requirement, requirement classifications, requirement characteristics, and requirement attributes; and most important, which of them are being considered for the development of this research work and the reason why.

It has been stated the importance of writing a "*good*" requirement, and its direct impact in future activities as verification; several Requirements Engineering process models have been illustrated, as well as their comparison in relation with ten desired quality characteristics they should have (linearity, support for changing requirements, iterative in nature, support for reverse engineering, risk assessment, criteria for application specific requirements elicitation techniques, requirements pre-processing, requirement prioritization, and effort estimation) (Batra & Bhatnagar, 2017); finally, requirement traceability has been defined.

Next chapter will present the state of the art on translating stakeholder needs into system requirements.

CHAPTER 3 Stakeholder Needs to Stakeholder Requirements

3.1 Introduction

This chapter reviews the state of the art about the translation of stakeholder needs into stakeholder requirements. Authors like Baudreau & Boulanger (2014), Ryan & Faulconbridge (2016), and SEBoK (2016) have done interesting propositions in this matter. The principal activities and tasks of this process are recognized; besides, this chapter describes and illustrates how several authors perform them.

3.2 State of the art

Figure 37 indicates the localization of the considered transformation process within the system life-cycle. As we can see, the process is positioned at the Concept stage, the beginning of the system life cycle. The stakeholder needs are the input for all the subsequent activities of the system life cycle. This is the reason why it is very important that the stakeholder needs are identified correctly.

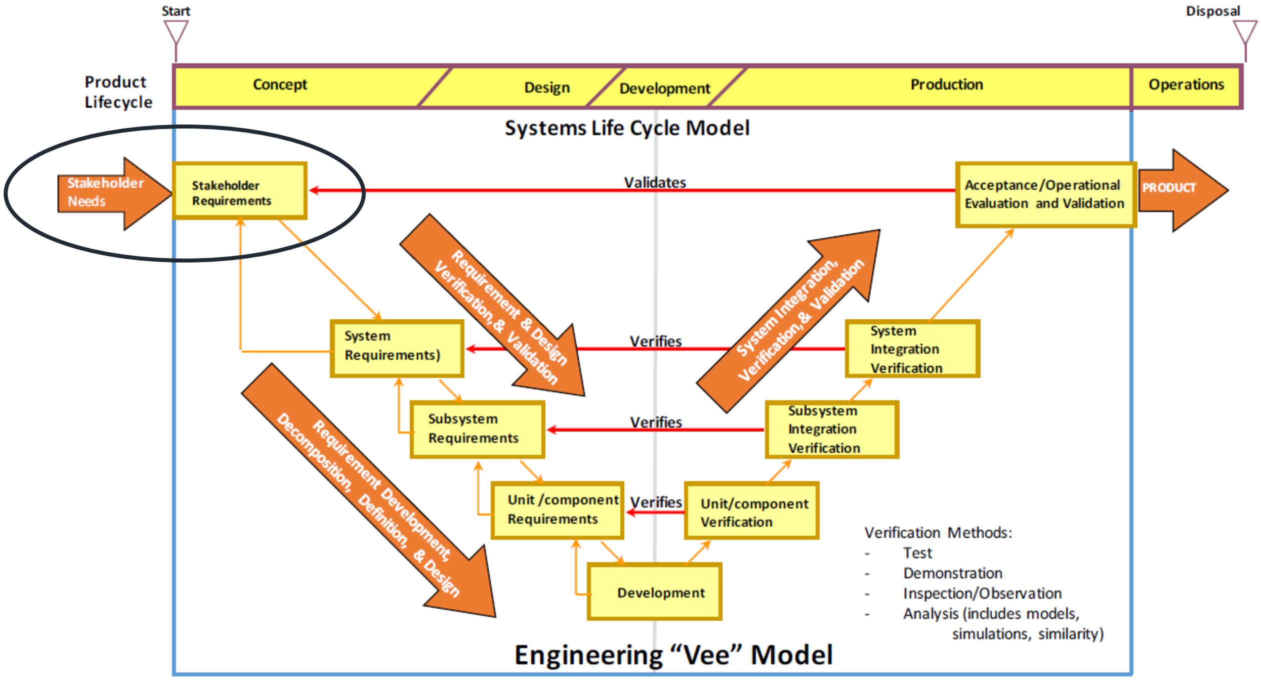


Figure 37 Localization of the Stakeholder Needs and Requirement definition process (adapted from Ryan & Wheatcraft, 2017)

According to several reference documents, such as Faisandier (2012), Ryan et al. (2015), or ISO 15288 (2015) to name a few, the objective of this process is to clearly and concisely elicit a set of stakeholder needs and expectations, and to transform them into stakeholder requirements, whose realization is verifiable in operation. Figure 38 shows how *real needs* are *perceived* and *expressed* by the stakeholders, to finally become *stakeholder requirements*.

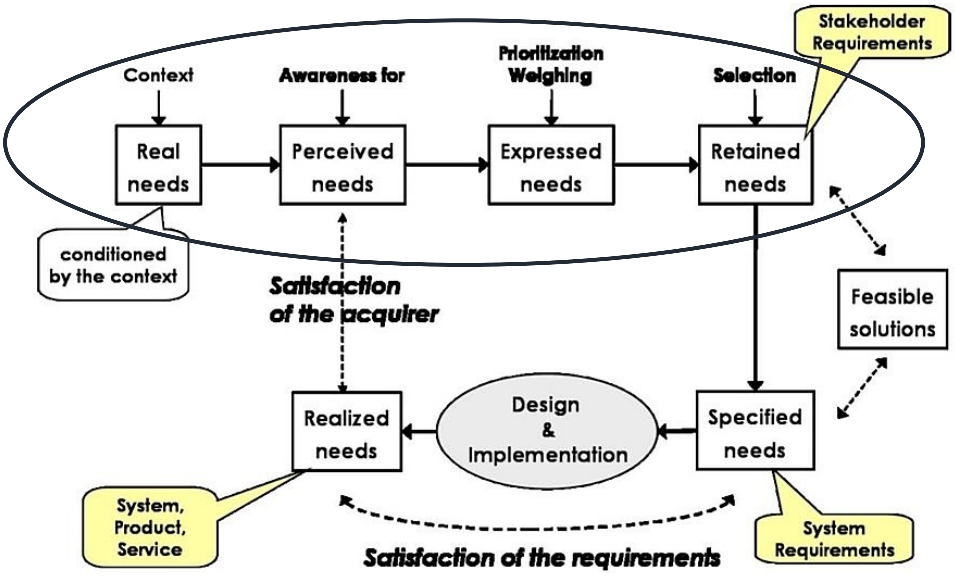


Figure 38 Cycle of needs (adapted from Faisandier, 2012, p.82)

To achieve this, the requirements engineer or analyst guides the stakeholders through a structured process of elicitation of needs. This can be done by applying elicitation techniques and methods that will be listed later.

SEBoK (2016) mentions that stakeholders initially have *vague* or *ambiguous* expectations that can be hardly used by Systems Engineering; the interdisciplinary analysis and design team must be careful to ensure that these expectations are defined in a clear and concise manner, to serve as a starting point for system definition. In this context, Faisandier (2012) and Ryan & Faulconbridge (2016) recommend modeling the context of the system using a higher-level system that encapsulates the System of Interest (SoI).

Faisandier (2012) proposes a process (Figure 39) for the definition of stakeholder requirements. The inputs for this process are the identified preliminary needs; later, the stakeholder needs are elicited and defined as stakeholder requirements, which are consolidated, verified, and validated, prior to be managed. These activities are carried out in sequence, nevertheless, iterations are necessary to obtain mature and consistent stakeholder requirements.

Faisandier (2014) states that the number of high-level elementary requirements of stakeholders should be between 30 and 80. A reduced number of high-level requirements will give more space to innovation while an extended number of requirements will narrowly orient towards a few numbers of solutions.

The set of stakeholder requirements will be the input for the process of system requirement definition, reviewed in Chapter 4.

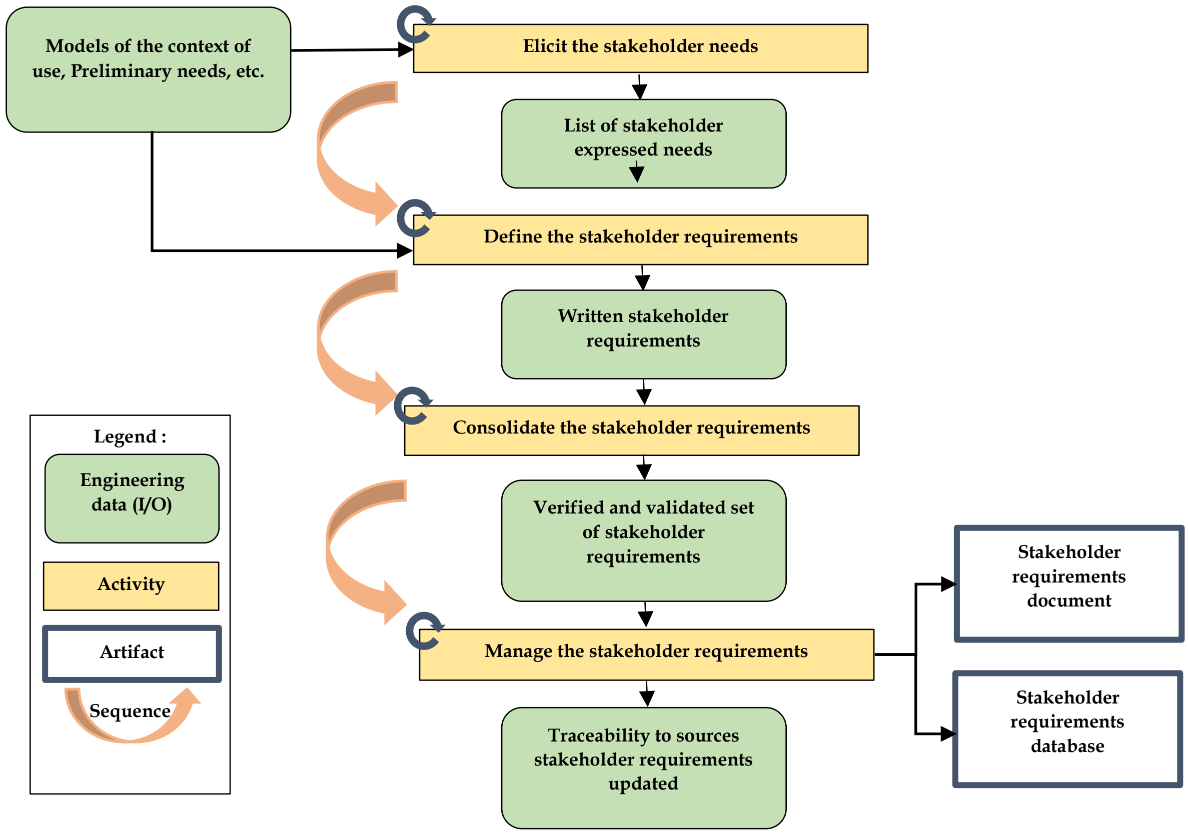


Figure 39 Activities of the Stakeholder Needs and Requirements Definition Process. (Faisandier, 2012, p.99)

The following of this section address twelve different activities identified in the literature of the process of transforming stakeholder needs into stakeholder requirements, and details each one of them specifically.

As stated earlier, this research work is conducted under Systems Engineering philosophy; that is the reason why in the Table 10 below are shown the activities proposed by Systems Engineering's authors to identify stakeholder needs and translate them into stakeholder requirements. The activities presented in Table 10 were identified after analyzing literature, and do not follow a strict order, this is because authors differ in their point of views.

Table 10 Activities to translate stakeholder needs into stakeholder requirements by author

Activities	Hull, Jackson, & Dick (2011)	Faisandier (2012, 2014)	Badreau & Boulanger (2014)	ISO 15288 (2015)	SEBoK (2016)	Ryan et al. (2015), Ryan and Faulconbridge (2016)	Aasheim & Zhao (2017)
Identify the stakeholders across the life cycle	•	•	•	•	•	•	•
Prioritize stakeholders							•
Elicit, capture, and consolidate the stakeholder needs, expectations, objectives, and constraints	•	•	•	•	•	•	•
Elicit stakeholders needs with the aid of requirement classifications						•	
Prioritize the retained stakeholder needs	•	•	•	•	•	•	•

Transform the prioritized and retained stakeholder needs into stakeholder requirements	•	•	•	•	•	•	•
Stakeholder requirement elaboration through decomposition and through derivation	•	•	•	•	•	•	•
Divide, classify and organize stakeholders requirements by type	•	•	•	•	•	•	•
Identify potential risks, threats, and hazards that could be generated by the stakeholder requirements		•	•	•	•	•	•
Verify the quality of stakeholder requirements	•	•	•	•	•	•	•
Validate the stakeholder requirements	•	•	•	•	•	•	•
Synthesize, record, and manage the stakeholder requirements and potential associated risks	•	•	•	•	•	•	•

Numerous authors such as Mitchell, Agle, & Wood (1997), Jepsen & Eskerod (2009) or Mazur (2012) for name a few, have worked overtime trying to improve the way these activities are conducted; in the following subsections the reader will find several author's contributions according to each activity shown in Table 10.

Activity: Identify the Stakeholders across the life cycle

The problem of stakeholder identification is not new.

In 1983 Freeman and Reed (Freeman & Reed, 1983) talked about the historical point of view towards stakeholders, starting from only considering shareowners, employees, customers, suppliers, lenders, and society, to the *rediscovery* of stakeholder analysis in the mid 1970s; the change from stakeholder *influence* to stakeholder *participation*; and in the late 1970s, when the environment became turbulent in the United States, the *stakeholder project* was conducted due to the need for a strategic management process.

In their article, they propose two processes to analyze the use of stakeholder concepts in strategy formulation processes: the "*Stakeholder Strategy Process*" and the "*Stakeholder Audit Process*". The first one is a systematic method for analyzing the relative importance of stakeholders, their cooperative potential, and their competitive threat; while the second one is a systematic method for identifying stakeholders and assessing the effectiveness of organizational strategies.

In Table 11 Freeman & Reed (1983) represent the *prevailing world view*: stockholders and directors have *formal or voting power*; customers, suppliers, and competitors have *economic power*; and government and special interest groups have *political power*."

Table 11 Prevailing world view (Freeman & Reed, 1983)

Power Stake	Formal or Voting	Economic	Political
Equity	Stockholders Directors Minority interests		
Economic	Debt holders	Customers Competitors Suppliers Foreign governments Unions	
Influencers			Consumer advocates Government Nader's Raiders Sierra Club Trade associations

On the other hand, in Table 12, Freeman & Reed (1983) represent the *real world* of stakeholder power, where a federal government agency is an influencer with great formal or economic power that should not be ignored.

Table 12 Real world stakeholder grid (Freeman & Reed, 1983)

Power Stake	Formal or Voting	Economic	Political
Equity	Stockholders Directors Minority interests		Dissident stockholders
Economic	Debt holders Unions	Suppliers Customers	Local governments Foreign Governments Consumer groups Unions
Influencers	Government SEC[1] Outside directors	EPA /OSHA	Nader's Raiders Government Trade associations

Some years later, Mitchell, Agle, & Wood (1997) propose a stakeholder identification based on three stakeholder attributes: power, legitimacy, and urgency (see Table 13).

According to these authors, each stakeholder may have one, two or three attributes; if one stakeholder has not *any* attribute, it means that she/he/it is a *non-stakeholder* and should not be considered anymore in the project. Thus, their theory helps to separate stakeholders from non-stakeholders

[1] Note: SEC means *Securities and Exchange Commission*, it is the U.S. federal agency for the regulation and control of financial markets; EPA means *United States Environmental Protection Agency*, an independent agency of the U.S. government, but controlled by the U.S. House of Representatives Committee on Science, Space and Technology; and OSHA means *Occupational Safety and Health Administration*, is a U.S. federal government agency whose mission is to prevent workplace injuries, illnesses and deaths, it issues regulations for occupational safety and health.

Table 13 Key constructs in the Theory on Stakeholder Identification and Salience (adapted from Mitchel, Agle, & Wood, 1997)

Construct	Definition
Stakeholder	Any group or individual who can affect or is affected by the achievement of the organization's objectives
Power	Relationships among social actors in which one social actor, A, can get another social actor, B, to do something that B would not have otherwise done
Bases	Coercive: force/threat Utilitarian: material/incentives Normative: symbolic influences
Legitimacy	A generalized perception or assumption that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values, beliefs, definitions
Bases	Individual Organizational Societal
Urgency	The degree to which stakeholder claims call for immediate attention
Bases	Time sensitivity: the degree to which managerial delay in attending to the claim or relationship is unacceptable to the stakeholder Criticality: the importance of the claim or the relationship to the stakeholder
Salience	The degree to which managers give priority to competing stakeholder claims

Then, the stakeholders are classified into 8 types. Figure 40 shows that a stakeholder may possess one, two, or three attributes; depending on what attributes the stakeholder has, it is typified as dormant, discretionary, dominant, dangerous, definitive, dependent, or demanding stakeholder.

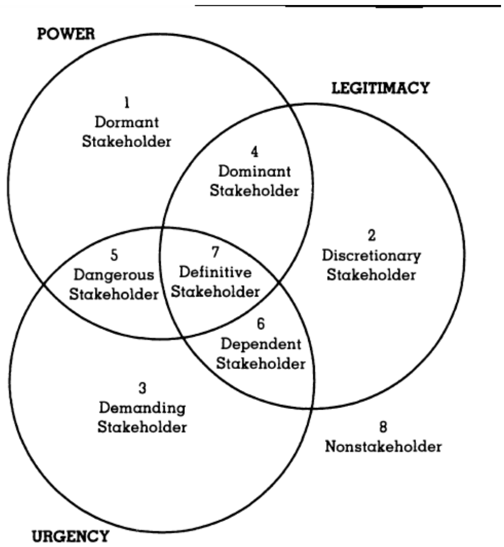


Figure 40 Stakeholder Typology (Mitchell, Agle, & Wood, 1997)

If the stakeholder has only one attribute, he is classified as a *latent stakeholder*; if the stakeholder has two of three attributes, he is classified as an *expectant* stakeholder; finally, if the stakeholder possess all three attributes, he is classified as a *definitive stakeholder* with the ability to influence the organization in the near future. Table 14 shows the stakeholder classifications based on how many attributes the stakeholders have, related to the stakeholder typologies.

Table 14 Stakeholder attributes, classification, and identification typology (Currie et al., 2009)

Attributes	Classification of Stakeholders	Identification Typology
Power	Latent Stakeholders	Dormant Stakeholders—while holding power, they lack legitimacy and urgency, therefore power is often unused
Legitimacy		Discretionary Stakeholder—holding no power or urgency to influence the organization
Urgency		Demanding Stakeholder—holding urgent claims yet lack the power or legitimacy to influence the organization
Power and Legitimacy	Expectant Stakeholders	Dominant Stakeholders—they have legitimate claims and the ability to act upon these claims by the power they hold
Power and Urgency		Dangerous Stakeholder—lack legitimacy yet has the power and urgency to influence the organization
Legitimacy and Urgency		Dependent Stakeholder—lack the power to carry out their urgent legitimate claims and therefore have to rely on others power to influence the organization
Power, Legitimacy and Urgency	Definitive Stakeholder	Definitive Stakeholder—Holding all three attributes the stakeholders has the ability to influence the organization in the immediate future

Mitchell, Agle, & Wood (1997) define *salience* as "*the degree to which managers give priority to competing stakeholder claims*"; the authors state that "*power gains authority through legitimacy, and it gains exercise through urgency*". Being congruent with this proposition, it can be said that identified stakeholders have different levels of importance to managers based on their level of salience. Figure 41 illustrates, for example, that a *dormant stakeholder* (who owns power attribute) has more salience than a *discretionary stakeholder* (who owns legitimacy attribute), and both of them have more salience than a *demanding stakeholder* (who owns urgency attribute).

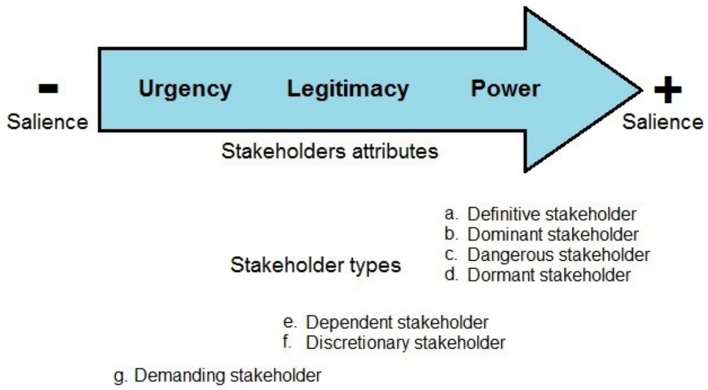


Figure 41 Stakeholder types according to their salience level

Sharp, Finkelstein & Galal (1999) in their article "Stakeholder Identification in the Requirements Engineering Process" propose an approach focused on the interactions among the different stakeholders types according to their *role,* the nature of their *relationship,* and *interaction*; their approach focuses on "*interactions between stakeholders rather than relationships* *between the system and the stakeholder, because they are easier to follow"* (Sharp, Finkelstein, & Galal, 1999). These authors identify the stakeholder *baseline* as:

a) **Users:** people, groups or companies who will interact with the system and control it directly, and those who will use the benefits (information, results, etc.) of the system.
b) **Developers:** are stakeholders in the requirements engineering process, but their stake in the final requirements specification, or indeed in the system itself; analysis and design team.
c) **Legislators:** professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for an operation that will affect the development and/or operation of the system.
d) **Decision makers:** within the development organization and the user organization, there will be decision-making structures that relate to the system under development; as managers or financial controllers.

Sharp, Finkelstein, & Galal (1999) observe and suggest:

- Identify exactly what happens in a workplace situation, observe existing practices, study work design for any new role, use participatory techniques to involve people, e.g. contextual inquiry
- Stakeholders may be internal to the team, internal to the organization, or external, or both
- Consider the full cycle of activities for each business, like a full financial year, or completed mission
- It is important to consider the full system life cycle when identifying baseline stakeholders

These authors define *three roles* that the baseline stakeholders play: *suppliers, clients,* and *satellites*. The *suppliers* provide information or supporting tasks; the *client's* processes or inspect the products of the baseline; the *satellites* interact with the baseline in a variety of ways: communicating, reading rules or guidelines, or searching information for example. The relation among *suppliers, clients,* and *satellites* can be seen in Figure 42.

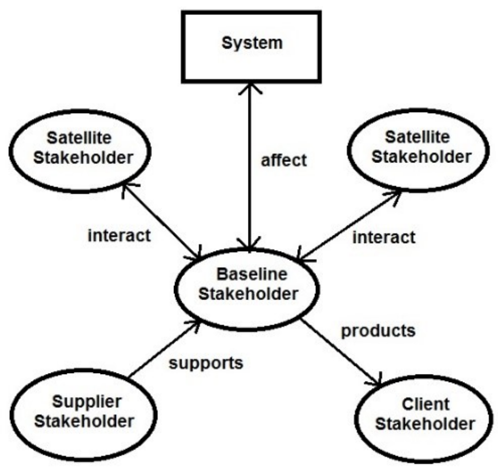


Figure 42 Main elements of stakeholder identification (Sharp, Finkelstein, & Galal, 1999)

Sharp, Finkelstein, & Galal (1999) propose the next process to identify the stakeholders systematically:

1. *Identify all specific roles within the baseline stakeholder group*
2. *Identify "supplier" stakeholders for each baseline role*
3. *Identify "client" stakeholders for each baseline role*
4. *Identify "satellite" stakeholders for each baseline role*
5. *Repeat steps 1 to 4 for each of the stakeholder groups identified in steps 2 to 4*

Glinz & Wieringa (2007) in their article "Stakeholders in Requirements Engineering" point out that the first activity of Requirements Engineering must be to determine *who* the stakeholders are, and *how important* they are. These authors focus on the identification of relevant stakeholders roles, though looking for persons or organizations who:

a) have an active interest in the system because they use or are involved in the process that the system will change
b) must manage, maintain, operate or introduce the system;
c) are involved in the project as an analyst, developer, architect, quality engineer or manager;
d) are responsible for the process or business that the system is supposed to support or automate;
e) have a financial interest;
f) are regulators of the system;
g) are affected by the system in a negative sense

The authors state that not all stakeholders are important in the same degree, so, they propose that stakeholder roles must be prioritized. This prioritization is based on assessing the incurred risk by ignoring or neglecting a stakeholder. Table 15 illustrates how risky is for the project to ignore the different stakeholder roles.

Table 15 Prioritization of stakeholder roles (Glinz & Wieringa, 2007)

Stakeholder role	System consequences if neglecting the stakeholder	Incurred risk
Critical	Might kill the project	high
Major role	Would impact significant and negatively the system	medium
Minor role	Would impact marginally the system	low

Jepsen & Eskerod (2009) explain in their article "Stakeholder analysis in projects: challenges in using current guidelines in the real world" that the list of stakeholders is generated by thinking of needed contributions to the project and who could/should provide these contributions or would be influenced by the project in other ways.

These authors propose that stakeholder analysis in projects requires the following activities:

a) Identification of the (*important*) stakeholders.
b) Characterization of the stakeholders pointing out their needed contributions, expectations concerning rewards for contributions, and power in relation to the project.
c) The decision about which strategy to use to influence each stakeholder.

Note: A *contribution* is a support needed from the stakeholders. "*Contributions are not just deliverances but also how stakeholders take part in the project work by participating in meetings and making decisions*" (Jepsen & Eskerod, 2009).

These authors suggest the application of tools like the "*stakeholder outline*" and "*stakeholder-commitment matrix*" respectively shown in Table 16 and Table 17. Table 16 illustrates important information related to every involved stakeholder to take into account during the entire process, like the *area of interest, contributions, expectations, power*, and *strategy*. To determine expectations and benefits desired by each stakeholder will allow the identification of where the *value* is (Jepsen & Eskerod, 2009; Mazur 2012; Bylund, Wolf, & Mazur, 2009; Oehmen, 2012; Schillo, Isabelle, & Shakiba, 2017).

Table 16 Stakeholder outline (Andersen et al., 2004, cited in Jepsen & Eskerod, 2009)

Stakeholder	Area of interest	Contributions	Expectations	Power	Strategy	Responsible

Mikkelsen & Riis (2007, cited in Jepsen & Eskerod, 2009) suggest that the stakeholder power may be assessed by the project team and the project manager, e.g. based on their knowledge about the formal and informal stakeholder roles and the organizational context. According to Jepsen & Eskerod (2009), level of power is recommended to be assessed in four categories: none, low, medium, and high, and it is important to be defined before eliciting needs, this activity consumes valuable time, and valuable stakeholders are the priority.

Table 17 shows how the stakeholder commitment may be managed (Jepsen & Eskerod, 2009) in order to avoid that stakeholder emotions, values, and beliefs cause the failure in system deployments during the Requirements Engineering processes (Ramos, Berry, & Carvalho, 2002; Berry, 2004; Ramos & Berry, 2005). These tools may be used for *stakeholder management*, a topic that will not be addressed in this research work.

Table 17 Stakeholder-commitment matrix (Mc Elroy, Mills, 2003, cited in Jepsen & Eskerod, 2009)

Stakeholder	Active Opposition	Passive Opposition	Neutral	Passive Support	Active Support
Suppliers			XO		
Top Management				X → → →	→ → → O
Colleagues in the permanent organization		X → → →	→ → →	→ → → O	
Grumbler				O ← ← ←	← ← ← X

X = current position, O = necessary/wanted position
*Example of the content filled in by the authors

Aasheim & Zhao (2017) proposes to prioritize the stakeholders through mapping them according to the power/interest grid illustrated in Figure 43; the author states that different stakeholders maintain different interests and influences in a project. Considering that it is nearly impossible to meet all stakeholder needs, it is relevant to identify those stakeholders with a high level of power and interest in order to *keep* them *close* and invite them to *cooperate* in the system development to increase the fulfillment of their needs. Other stakeholders can only be monitored, informed, or kept satisfied.

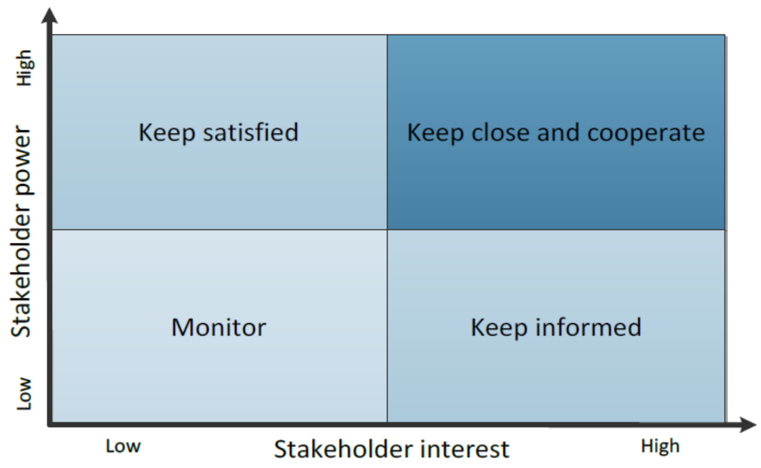


Figure 43 Stakeholder grid (Aasheim & Zao, 2017)

Bylund, Wolf, & Mazur (2009) make interesting questions as "*are all customers equally important? Are there similarities between different types of customers?*" To answer these questions, Mazur (2012) assures that the key customers may and must be prioritized applying the Analytical Hierarchy Process (AHP).

Note: in the domain of Quality are called *customers* those who in Systems Engineering are *stakeholders*.

Table 18 is an example of Customers Segments Table (CST) proposed by Mazur (2012) to record stakeholder information. The CST is based on the tool *5 W's and How* (*Who, What, Where, When,* and *Why*).

Table 18 Example of Customer segments table (Mazur, 2012)

Customer Segment	Who uses process	What disease condition	What is process used for	Where	When	Why	How
Pediatric pt	Pt and family	spinabifida	improve interaction between visting physicians	CMH main campus	daytime ambulatory clinic	visiting physicians do not have sufficient time to interact	develop new tools to better evaluate treatment options
Visiting physicians	Provider	mild traumatic brain injury				Share medical records from urban to rural facility members	
Nursing staff		muscle and nerve					

Mazur (2012) suggests that the *customers* should be defined based on the interactions with the proposed product, process or service.

Faisandier (2012) recommends paying special attention to being exhaustive, in other words, identify all the stakeholders and interview them. Bylund, Wolf, & Mazur (2009) suggest to interview them *in situ*, in the "crime scene" to learn about the company culture. Schillo, Isabelle, & Shakiba (2017) states that with the aid of QFD it is possible to do systematically the consideration of all stakeholders.

Turner & Zolin (2012) propose 8 stakeholders types:

1. **Owner or investor:** who pays, by, pay for its operation, and obtain revenue
2. **Project executive or project sponsor:** prior to the Project, identify the need for a new asset and the potential benefit it will bring

3. **Consumers:** buy the product the new assets produces
4. **Operators/users:** who operate the asset on behalf of the owner
5. **Project manager and Project team**
6. **Senior supplier** (design and/or management): senior management in the lead contractor
7. **Other suppliers:** people or groups who provide goods, materials; Works or services
8. **Public:** The public concerned with environmental and social impacts of the system, they will want to know how their taxes have been spent

On its side, SEBoK (2016) suggests that the *product integration teams* (stakeholders) include strategic planners, business managers, financial managers, market managers, quality assurance managers, customer representatives, and end-users, as well as other disciplines required for acquired products. SEBoK (2016) mentions that these stakeholders should be involved early in the analysis, in order to cooperate at all stages of the life cycle in specifying requirements, verifying that the requirements are met, and validating that the products produced provide needed capabilities.

Unfortunately, errors may show up when identifying stakeholders through the system life cycle. Faisandier (2012) and SEBoK (2016) consider as grave errors not considering the operator/user as a stakeholder and forgetting certain categories of stakeholders such as persons who do not want the system or are malevolent. Ramos Santos et al. (1998, 2002, & 2004 cited in Barry, 2004) report several failed informatics technology system deployments because of user sabotage or user fears.

Once that the stakeholders have been identified, in some cases prioritized, considered, and invited to participate in the development of the new system, it is time to elicit their needs.

Activity: Elicit, capture, or consolidate the stakeholder needs, expectations, objectives, and constraints defined by mission, purpose, and objectives

According to Nuseibeh & Easterbrook (2000), the elicitation of needs help to delimit the system and to establish its boundaries.

Hickey & Davis (2003) conducted interesting research work to discover how experts elicit needs. "*Many practitioners are looking for a simple recipe for success … that will solve all their elicitation problems … However, consensus exists that one elicitation technique cannot work for all situations*" (Hickey & Davis, 2003). Through their findings, it can be seen that elicitation process is performed in a wide variety of situations, with a combination of participants, problem domains, solution domains, organizational contexts, and in an extensive variety of ways.

Several authors recommend the use of methods and techniques for elicitation and identification of stakeholder needs, expectations, and requirements; these methods and techniques are shown in the following Table 19.

Table 19 Techniques or Methods recommended by different authors for the elicitation of stakeholder needs.

Technique or Method	Hickey & Davis (2003)	Blanchard & Fabrycky, (2006)	Faisandier (2012)	SEBoK (2016)	Ryan et al. (2015) Ryan and Faulconbridge (2016)
Structured brainstorming workshops	•		•	•	•
Interviews and questionnaires	•		•	•	•
Technical documentation review			•		
Organizational analysis techniques e. g. FODA			•		
Market-Driven Product Definition – MDPD			•		
Quality Function Deployment – QFD		•	•	•	
Goals Method			•		
Problem-solving sessions	•				•
Use cases and operational scenarios				•	•
Simulation, models, and prototypes	•	•		•	•
Time and motion studies					•
Participation in work activities					•
Observation of the system's organizational and political environment	•				•
Technical documentation review				•	•
Market analysis					•
Competitive system assessment	•			•	•
Reverse engineering					•
Benchmark processes and systems					•
Use case diagrams				•	
Activity diagrams				•	
Feedback from verification and validation processes				•	
Functional flow block diagrams				•	
Input-output Matrix		•			
Value engineering		•			
Focus Group				•	
Delphi Techniques				•	
Statistical data analysis	•	•			
Trend analysis		•			
Matrix analysis		•			
Parametric Analysis		•			
Tradeoffs		•			
Collaborative sessions	•				
Team building	•				
Ethnography	•				
Issues lists	•				
Requirements categorization	•				•
Role-playing	•				
Extreme programming	•				

According to Table 19, the most popular techniques are brainstorming, interviews and questionnaires, QFD, simulation, models, and prototypes. Brahm & Kleiner (1996) point out the advantages and disadvantages of some of the above techniques and add others. Table 20 contains comparative advantages and drawbacks of brainstorming, brain-writing, quality circles, and nominal group technique.

Table 20 Comparative: advantages and disadvantages of brainstorming, brain-writing, quality circles, and nominal group technique (Brahm & Kleiner, 1996)

Techniques or Methods	Advantages	Disadvantages
Brainstorming	An enormous number of ideas are generated	It produces ideas without screening them. Many of the ideas may not be "quality ideas". Effective solutions to problems usually have been screened, tested, and evaluated. Useful when there is a small group of individuals, plenty of time, minimal differences among status group members, and a need exists to verbally discuss ideas with others.
Brain-writing	This technique will produce more ideas than brainstorming A facilitator is not needed People concerns of expressing themselves orally and in front of a group is eliminated	Expressing ideas in writing is not comfortable for certain people Only highly useful for very large groups, when there is little time available, status differences are equalized, and there is no need for verbal interaction.
Quality circles	This technique produces Employee involvement, commitment and cohesiveness, and higher quality production	The supervisor of the method serves only as a guide rather than a boss One the group recommends the decision, the supervisor must either accept or reject the idea and held the consequences This technique works only with top management support
Nominal group technique (NGT)	It is a time-saving technique that: minimize differences and ensuring relatively equal participation, decreases tension and hostility, and produces a large number of ideas and provides a sense of closure. It is excellent when used in meetings that are concerned with judgmental decision making	The method lacks flexibility by only being able to deal with one problem at a time. There must be a certain amount of conformity on the part of the members involved in NGT. Another disadvantage is the amount of time needed to prepare for the activity. There is no spontaneity involved with this method. Facilities must be arranged and carefully planned.

Brahm & Kleiner (1996) remark that the Nominal Group Technique (NGT) *"is a procedure that combines both features of brainstorming and brain-writing to produce a highly effective group decision-making process. NGT process involves the following six basic steps:*

1. *silent generation of ideas in writing*
2. *recording of ideas*
3. *discussion for clarification*
4. *a preliminary vote on item importance*
5. *discussion of the preliminary vote*
6. *final vote"*

Hull, Jackson, & Dick (2011) suggest that the workshops are an excellent way of rapidly eliciting and capturing requirements; the workshop should be structured, but also iterative. Figure 44 illustrates that the stakeholders should be trained to understand what is expected from them, and the concepts of *stakeholder*, *use scenarios*, and *capability requirements* for example.

The process starts by splitting the participants into teams and allow them to create scenarios for the system of interest; next, make any required change(s) to the scenarios and continue with the requirement elicitation based on them. As soon as possible, based on draft requirements, encourage criticism and interaction among stakeholders; the interaction will add value to the requirements. The authors recommend to edit requirements online and work in small groups to be more productive; once the set of requirements is complete, all participants can review them together. This way, the set of requirements can be produced in three or four days. "*It is vital that all stakeholder groups are represented and that they are empowered to make decisions*" (Hull, Jackson, & Dick, 2011).

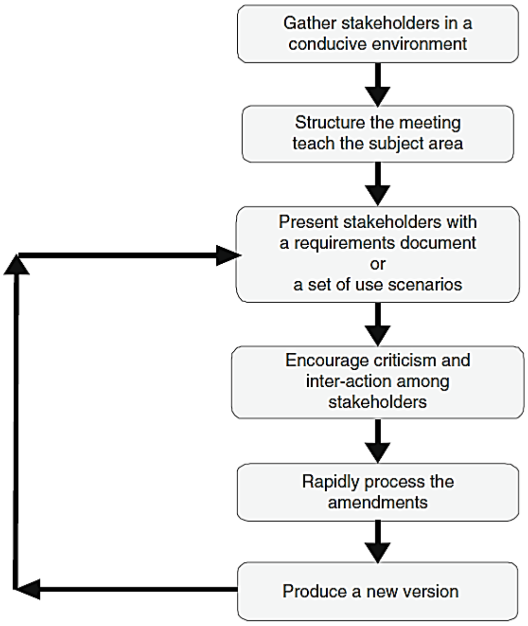


Figure 44 Workshops for requirements capture (Hull, Jackson, & Dick, 2011)

Akbayrak (2000) presents a comparison between interviews and questionnaires, stating that "*in some cases, the combination of both two techniques might be the best way to provide a powerful research strategy instead of trusting on only one technique*"; in the same perspective, Alshenqeeti (2014) adds that "*interviewing is a powerful way of getting insights into interviewee's perceptions, it can go hand in hand with other methods providing in-depth information about participants' inner values and beliefs*". Table 21 illustrates how the disadvantages of interviews are covered by the advantages of questionnaires, and vice-versa.

Table 21 Summary of advantages and disadvantages of interviews and questionnaires (Akbayrak, 2000)

Criteria	**Interviews**		**Questionnaires**	
	Advantages	Disadvantages	Advantages	Disadvantages
Access to information	✔			✗
Anonymity		✗	✔	
Application Skill		✗	✔	
Bias		✗	✔	
Confidentiality		✗	✔	
Cost		✗	✔	
Data Analysis		✗	✔	
Flexibility	✔			✗
Reliability	✔			✗
Response Rate	✔			✗
Sample size and Sampling		✗	✔	
Time		✗	✔	
Validity	✔			✗

Meaning of symbols: ✔ Advantage ✗ Disadvantage

In 2000, Bouchereau & Rowlands (2000) studied the benefits and drawbacks of QFD, their findings are shown in Table 22.

Table 22 Benefits and drawbacks of QFD. (Bouchereau & Rowlands, 2000)

Benefits	Drawbacks
Customer oriented	Ambiguity in the VOC[2]
Brings together large amounts of verbal data	Need to input and analyze large amounts of subjective data
Bring together multi-functional teams	QFD development records are rarely kept
Reduces development time by 50 percent and reduces start-up and engineering cost by 30 percent	Manual input of customer survey into the house of quality (HoQ) is time-consuming and difficult
Helps design quality into the products at the design stage	QFD analyzes often stop after the first HoQ, so links between the four QFD phases are broken
Organizes data in a logical way	The HoQ can become very large and complex
QFD is used not only for products but for processes and services as well	Setting target values in the HoQ method is imprecise
Strengthens the good relationship between customer and company	Strength of relationship is ill-defined
Improves customer satisfaction	QFD is a qualitative method

Bouchereau & Rowlands (2000) state that QFD is a management tool to help analysis and design teams focus on the needs of the customers throughout the total development cycle of a system or process. This tool helps to develop more customer-oriented, higher-quality products. Nevertheless, it is not simple to use. Their article outlines how techniques such as *fuzzy logic*, *artificial neural networks*, and the *Taguchi method* can be combined with QFD to resolve some of the identified drawbacks and proposes a synergy between QFD and these three methods.

See Figure 45. Bylund, Wolf, & Mazur (2009) apply Blitz QFD for the process of elicitation of needs; they suggest to make on-site visits to stakeholders (*gemba*), divide them into groups according to their level of abstraction of thinking, and develop different diagrams that will support the elicitation of needs (Customer Segments Table or CST, number 1 in Figure 45). With the findings of *gemba*, the analysis and design team may build the Customer Process Model (CPM, number 2 in Figure 45) and the Gemba Visit Table (GVT, number 3 in Figure 45). This model is a representation of the stakeholder processes; it will help the understanding and gathering of information *in situ*. During the *gemba*, together the analysis and design team with the stakeholder should do failure modes and failure effects analysis to get valuable information.

The reason for proposing a visit to the "*crime scene*" or *gemba* (Bylund, Wolf, & Mazur, 2009; Alshenqeeti, 2014) is that they can become observers, not just interviewers, and thus enrich their findings, better understand the culture of stakeholders, investigate their external behavior, and internal beliefs. Bylund, Wolf, & Mazur (2009) and Mazur (2012) suggest applying one template called Customer Voice Table (CVT) (number 4 in Figure 45) to translate any form of data from the CPM and the GVT into customer needs. Immediately, these authors suggest making a structure of the needs listed in the CVT according to their affinity, they suggest the application of the Affinity Diagram (number 5 in Figure 45).

[2] Note: VOC means voice of the client.

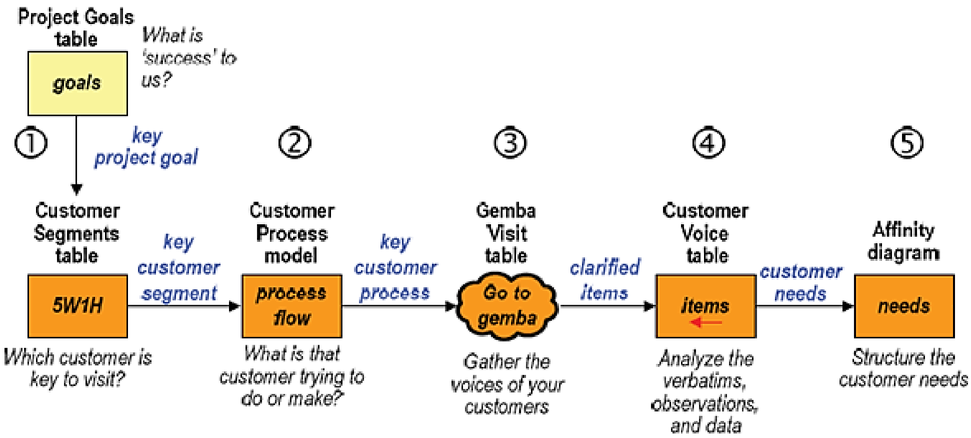


Figure 45 Custom tailored Blitz QFD ® Process (adapted from Bylund, Wolf, & Mazur, 2009)

Mazur, Terniko, & Ziltner (2017) proposes the application of the Kansei Engineering Tool as a new product development method focused on unspoken customer needs and emotional branding, and that can be applied in the QFD framework.

In the domain of Systems Engineering, several authors such as Faisandier (2012) and Ryan & Faulconbridge (2016) propose to initiate the process of need elicitation starting from the mission of the system and hierarchizing its main functions. For example, Figure 46 illustrates that the mission of the system is supported by Function 1 and Function 2, while Function 1 is based on Function 1.1, Function 1.2, and Function 1.3.

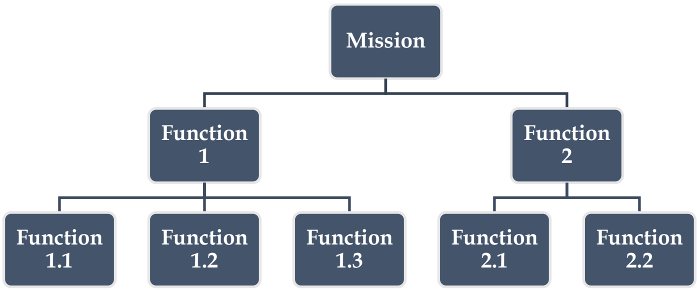


Figure 46 Hierarchy of requirements (Ryan & Faulconbridge, 2016)

Then, Faisandier (2012) goes deeper and suggests following the process with what he calls "intended use of the system", which includes:

- *"**Purpose:** why the system exists? What is the usage and relevance of the system in this context of use?*
- ***The intended use (mission):** What it does? Which transformation does it perform to achieve its purpose?*
- ***Objectives:** Which are the main performances (quantified) that the system must satisfy so that the purpose is achieved? How many inputs does the system transform? How speed? "* (Faisandier 2012)

According to Van Lamsweerde (2000), these identified objectives of the system can be contrasted to the possible system operational scenarios in order to identify contradictions.

Sadig & Sahraoui (2017) in their article " Requirements Engineering Practice in Developing Countries, Elicitation and Traceability Processes " suggest fifteen core information type "*that* *must be addressed*" during the requirement elicitation phase. Note that in Table 23 are mixed both, system and project requirements.

Table 23 Fifteen core information types (Sadig & Sahraoui, 2017)

No.	Title	Description
1	Project	Problem mission, vision, context, and scope of the Project
2	Deliverable	The desired result of the process, its audience, objectives, and overview
3	System	Background, perspective, context, and scope of the system
4	Objectives	Objectives of the business with respect to the Project and system
5	Assumptions	Underlying assumptions upon which the Project and system are based
6	Constraints	Constraints that must be applied to the Project and system
7	Environment	Social and physical environmental characteristics of the Project and system
8	Opportunities	Possible opportunities for improvements to be delivered by the system
9	Challenges	Possible challenges which may be encountered during the Project
10	Risks	Potential risks to both, the Project and the system
11	Stakeholders	Stakeholders in the Project, and sources of system information
12	Processes	Detailed work process which the system must support
13	Functional	Functional aspects which must be provided by the system
14	Non-functional	Non-functional aspects which must be provided by the system
15	Implementation	Implementation details relating to the system including design solutions

Authors, like Sommerville, Sawyer, & Viller (1998), Ryan et al. (2015) and Ryan & Faulconbridge (2016), propose to support elicitation with requirements classifications.

The elicitation and development of stakeholder requirements are complex activities because the analysis and design team must understand the business, application domain, specific problem, system needs and constraints, project acquisition and management, requirements engineering and systems engineering, as well as the technology and engineering involved (Ryan & Faulconbridge, 2016).

Mokammel et al. (2018) explain that the elicitation of needs could be performed from various sources: emails, regulation documents, meeting minutes, etc.; inevitably, these needs can be expressed in the form of text, drawings, tables, models, figures, etc. (Faisandier, 2012; SEBoK, 2016). Mokammel et al. (2018) add that in this stage it is necessary that different software platforms and forms may interact, this consideration should be seen as a *key* issue. They propose the use of common Requirement Interchange Format (ReqIF[3]) to solve this challenge.

Activity: Prioritize the stakeholder needs

Faisandier (2012) proposes to sort the stakeholder needs and keep the essential and valuable ones. Davis (2003 cited in Barry 2004) ranks requirements in the following five categories: 1. Absolutely essential, 2. Essential, 3. Important, 4. Nice, and 5. Fluff.

In the QFD frame (Bylund, Wolf, & Mazur, 2009; Mazur, 2012), the hierarchy of stakeholder needs are truly reflected in the Hierarchy Diagram (HD) (number 6 in Figure 47) and will be the basis for the prioritization of needs (number 7 in Figure 47). Together, the Affinity Diagram and HD structure will help to discover missing and unspoken stakeholder needs.

[3] Note: The ReqIF is an XML file format applied to easy the exchange requirements among companies.

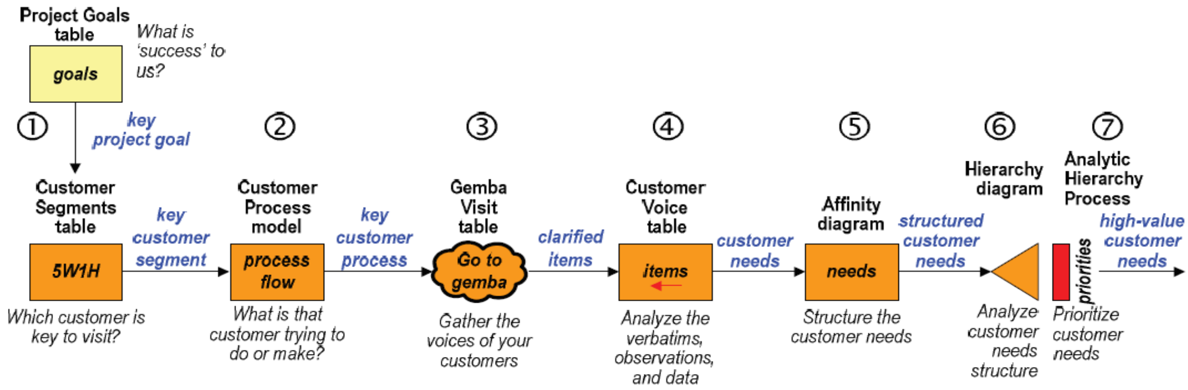


Figure 47 Custom tailored Blitz QFD ® Process (adapted from Bylund, Wolf, & Mazur, 2009)

The QFD Methodologies (Hauser & Clausing, 1988) and Blitz QFD (Bylund, Wolf, & Mazur 2009; Mazur, 2012; and Schillo, Isabelle, & Shakiba, 2017) allow prioritizing stakeholder needs through the application of the Analytical Hierarchy Process (AHP), developed by Saaty (2007), to detect any inconsistencies in the prioritization.

Danesh, Ryan, & Abbasi (2015) express that there are more than 100 Multi-Criteria Decision Making (MCDM) techniques. All of them has advantages and disadvantages and they need to be selected based on the specific situation. Nevertheless, when there are complex independencies and a large number of criteria or alternatives, it is recommended to manage them through a hierarchical format, like the AHP method.

The AHP approach employs a pairwise analysis for arranging preferences. First, the stakeholders built the pairwise comparison matrix, using the intensity scales shown in Table 24.

Table 24 Comparative judgment table (Danesh, Ryan, & Abbasi, 2015)

	Intensity Scale	
Less important than	Extremely less important	1/9
		1/8
	Very strong less important	1/7
		1/6
	Strongly less important	1/5
		1/4
	Moderately less important	1/3
		1/2
	Equal Importance	1
More important than		2
	Moderately more important	3
		4
	Strongly more important	5
		6
	Very strong more important	7
		8
	Extremely more important	9

To better illustrate, it is shown an example in Table 25. The resulting pairwise comparison matrix among factors as time, cost, quality, risk, and work, health and safety (WHS).

Table 25 Pairwise comparison matrix (Danesh, Ryan, & Abbasi, 2015)

Factor	Time	Cost	Quality	Risk	WHS
Time	1	3	3	5	2
Cost	1/3	1	1	1	1
Quality	1/3	1	1	1	2
Risk	1/5	1	1	1	1
WHS	1/2	1	1/2	1	1
Total	**2.367**	**7.000**	**6.500**	**9.000**	**7.000**

Where the interpretation of factor *time* related to factor *cost* is:

- Time is *moderately **more** important* than cost = 3
- Cost is *moderately **less** important* than time = 1/3

Later, the comparison matrix needs to be normalized. This can be done through some calculations; for example, the first column calculations are shown in Table 26.

Table 26 Calculations for the first column

Factor	Time	
Time	**0.423**	**= 1 / 2.367**
Cost	0.141	**= 1/3 / 2.367**
Quality	0.141	**= 1/3 / 2.367**
Risk	0.085	**= 1/5 / 2.367**
WHS	0.211	**= 1/2 / 2.367**
Total	**1.000**	**= 2.367 / 2.367**

The normalized matrix of the example is shown in Table 27.

Table 27 Parameter weights A (Danesh, Ryan, & Abbasi, 2015)

Factor	Time	Cost	Quality	Risk	WHS
Time	**0.423**	0.429	0.462	0.556	0.286
Cost	0.141	**0.143**	0.154	0.111	0.143
Quality	0.141	0.143	**0.154**	0.111	0.286
Risk	0.085	0.143	0.154	**0.111**	0.143
WHS	0.211	0.143	0.077	0.111	**0.143**
Total	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Once the matrix is normalized, it can be calculated the *total* by line and the *weight vector*; the weight vector represents the prioritization value; the highest value means the *most important* factor.

For example, to calculate the *total* (addition of all line values) and *weight vector* (average of all line values) for the factor *time*:

Factor	Time	Cost	Quality	Risk	WHS	Total (Factors) Adition	Weight Vector Average
Time	**0.423**	0.429	0.462	0.556	0.286	2.154	0.431

Table 28 shows the *total (factors), the weight vector,* and the *percentage* or relative importance among the factors.

Table 28 Parameter weights B (Danesh, Ryan, & Abbasi, 2015)

Factor	Total (Factors)	Weight Vector	%
Time	2.154	0.431	**43.08%**
Cost	0.692	0.138	**13.83%**
Quality	0.834	0.167	**16.69%**
Risk	0.635	0.127	**12.70%**
WHS	0.685	0.137	**13.70%**

Total = 100%

These results can be interpreted as follows:

- The most important requirement is *time*, with a weight of 43.08%.

Note that this objective data may be useful later, for validation purposes. If this requirement is fulfilled, the system meets 43.08% of stakeholder needs.

That is the importance of prioritizing stakeholder needs, the analysis and design team can focus the resources, always limited, in order to satisfy the most important stakeholder needs. Applying the AHP method allows to quantify stakeholder satisfaction, this numerical value is objective and measurable.

Finally, the AHP approach requires to verify the *consistency* among the stakeholder responses when filling the pairwise comparison matrix. Consistency analysis is done through the calculation of the *consistency ratio* (CR). The CR of less than 0.1 or even slightly above 0.1 is regarded as sufficient (Danesh, Ryan, & Abbasi, 2015). See (1).

$$CR = \frac{\lambda_{\max} - N}{(N-1)RI} \quad (1)$$

Where: $\lambda_{\max}$ = maximal eigenvalue
N = dimension of the matrix
RI = random index, see Table 29

Table 29 Random Index Form (Danesh, Ryan, & Abbasi, 2015)

n	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.46	1.49

To obtain $\lambda_{\max}$ it is necessary to obtain the consistency measure of every factor. Table 30 illustrates the example used here.

Table 30 Consistency measure (Danesh, Ryan, & Abbasi, 2015)

Consistency Measure	
Time	5.236
Cost	5.154
Quality	5.093
Risk	5.159
WHS	5.118
λ_{max}	**5.152**

By substituting the values from the example in (1) it can be seen that the *consistency ratio* value (CR) is 0.034, it is, less than 0.1 and in consequence, the paired comparison was conducted in a consistent manner resulting in a trustworthy prioritization of factors:

$$\lambda_{max} = 5.152$$

$$CR = \frac{\lambda_{max} - N}{(N-1)RI} = \frac{5.152 - 5}{(5-1)1.12} = 0.034$$

CR= 3%

Consistency= OK

Bylund, Wolf, & Mazur (2009) state that a shared knowledge or the aim of the project align and motivate the people involved in the project, in other words, a shared knowledge throughout the organization of what the customer need and the understanding of the conditions of use aligns the development, the introduction, and the sales efforts.

Many researchers had developed their own methods to prioritize elements. Nevertheless, Khan et al. (2016) describe AHP as *"the best method"* to prioritize, and express the need for tool support due to a large number of pair comparisons to be done.

In 2016, Khan et al. (2016) define the analytic network process (ANP) as "*a multi-criterion theory of measurement used to obtain relative priority scales of absolute numbers from individual judgments that also belong to the fundamental scale of absolute numbers... ANP has a lot of applications almost in every field"*. The authors used AHP as the basis, in addition, ANP considers interdependencies amongst the criteria and alternatives. The advantages and disadvantages of ANP method are shown in Table 31.

Table 31 Advantages and disadvantages of ANP method. (Khan et al., 2016)

Advantages	Disadvantages / Limitations
ANP can prioritize both dependent and independent requirements	Prioritization process in ANP is complex
ANP provides reliable and fault tolerant results	Tool support is needed to minimize the complexity and time consumption while prioritizing requirements
ANP gives consistent results	
ANP gives results on ratio scale which further improves the prioritization process	

These authors conducted a very interesting study that compares the ANP against other methods; this evaluation was objective (the ease of use and the reliability of results) and subjective (the required number of decisions, the total time consumed, and the time consumed par decision). The conclusion identified that "*more research and hard work is needed in the field of requirements prioritization to improve the performance of ANP*" (Khan et al., 2016). See Table 32 and Table 33.

Table 32 Subjective measures after evaluation of requirements. Comparative among binary search tree, AHP, hierarchy AHP, spanning tree matrix, Priority group and bubble sort prioritization techniques (Khan et al., 2016)

Evaluation criteria	ANP	AHP	Hierarchy AHP	Spanning tree	Bubble sort	Binary search	Priority groups
Ease of use	4	3	5	4	1	6	3
Reliability of results	1	1	3	6	4	5	4

Table 33 Objective measures after evaluation of requirements. Comparative among binary search tree, AHP, hierarchy AHP, spanning tree matrix, Priority group and bubble sort prioritization techniques (Khan et al., 2016)

Evaluation criteria	ANP	AHP	Hierarchy AHP	Spanning tree	Bubble sort	Binary search	Priority groups
Required number of decisions	140	45	15	9	45	27	20
Total time consumption (ordinal scale 1-6)	6	5	3	1	2	4	3
Time consumption per decision (ordinal scale 1-6)	3	2	5	6	1	6	3

Rokou (2009) proposes the ANP SOLVER, free software for academic purposes to ease the implementation of ANP.

The ANP SOLVER is free and available at http://kkir.simor.ntua.gr/anpsolver.html

Once that stakeholder needs are prioritized, it is time to translate them into stakeholder requirements (Faisandier, 2012).

Activity: Transform the prioritized and retained stakeholder needs into stakeholder requirements

According to Sommerville, Sawyer & Viller (1998), Hickey & Davis (2003), Ryan et al. (2015) and Ryan & Faulconbridge (2016) the categories of requirements can be used when transforming needs into requirements; then, the focus attention is on various perspectives of the system and this makes sure that all the areas have been addressed resulting in a complete set of requirements.

SEBoK (2016) suggests that this transformation should be guided by a well-defined, repeatable, rigorous, and documented process, and recommends applying tools such as functional flow diagrams, timeline analysis, N2 diagrams, design reference missions, modeling and simulations, movies, pictures, states and modes analysis, fault tree analysis, failure modes and effects analysis (FMEA), and trade studies.

Chapter 2 described the state of the art of how to express requirements and the expected quality characteristics they should possess. Nevertheless, it can be added some additional recommendations.

Common mistakes when writing stakeholder requirements

Faisandier (2012) and SEBoK (2016) agree on which are the most serious errors during the definition of stakeholder needs and requirements:

- ✗ Operator role not considered as a stakeholder
- ✗ Exchanges with external objects forgotten
- ✗ Physical connections with external objects forgotten
- ✗ Forgotten stakeholders such as persons who don't want the system or malevolent
- ✗ Lack of domain or environment knowledge

Proven practices when writing stakeholder requirements

Faisandier (2012) and SEBoK (2016) mention which practices have been tested and should be replicated during the definition of stakeholder needs and requirements:

- ✓ Involve stakeholders early in the analysis
- ✓ Presence of rationale for each stakeholder requirement
- ✓ Analyze sources of stakeholders requirements before starting the definition of the system requirements
- ✓ Use of modeling techniques
- ✓ Use of requirements managing tool to trace linkages and to record the source of each stakeholder requirement

Faisandier (2012) suggests a certain type of wording to express stakeholder requirements and can be consulted directly in his book. The author strongly recommends:

1. Model the system of interest (SoI) from a higher-level system to identify and define its context (services, functional interfaces, and physical interfaces)

2. Use language and synonym dictionaries. Semantics is the key to the correct expression of requirements

Szejka et al. (2014) propose in their article "*Requirements interoperability method to support integrated product development*" a conceptual method for requirements interoperation in Integrated Product Development Engineering, that is, a method to transform *informal requirements* into *formal requirements*. Figure 48 illustrates how the stakeholders express their requirements, which are analyzed and translated into *product requirements*; later, through formal logic, they are traduced into *product requirements* and stored in repositories to future re-use. Requirement formalization is done through fact-oriented modeling (letter A in Figure 48), object role modelling (letter B in Figure 48), verbalization (letter C in Figure 48), Verification (letter D in Figure 48), logical modelling (letter E in Figure 48), and common logic (letter F in Figure 48).

NOTE: After reading and reflecting on the article by Szejka et al. (2014), and due to the difference in languages between the different disciplines of knowledge, it is concluded that:

Szejka et al. (2014) article language		***Systems Engineering language***
Stakeholder requirements	*means*	stakeholders needs
Product requirements (Informal)	*means*	stakeholders requirements
Product requirements (Formal)	*means*	system requirements.

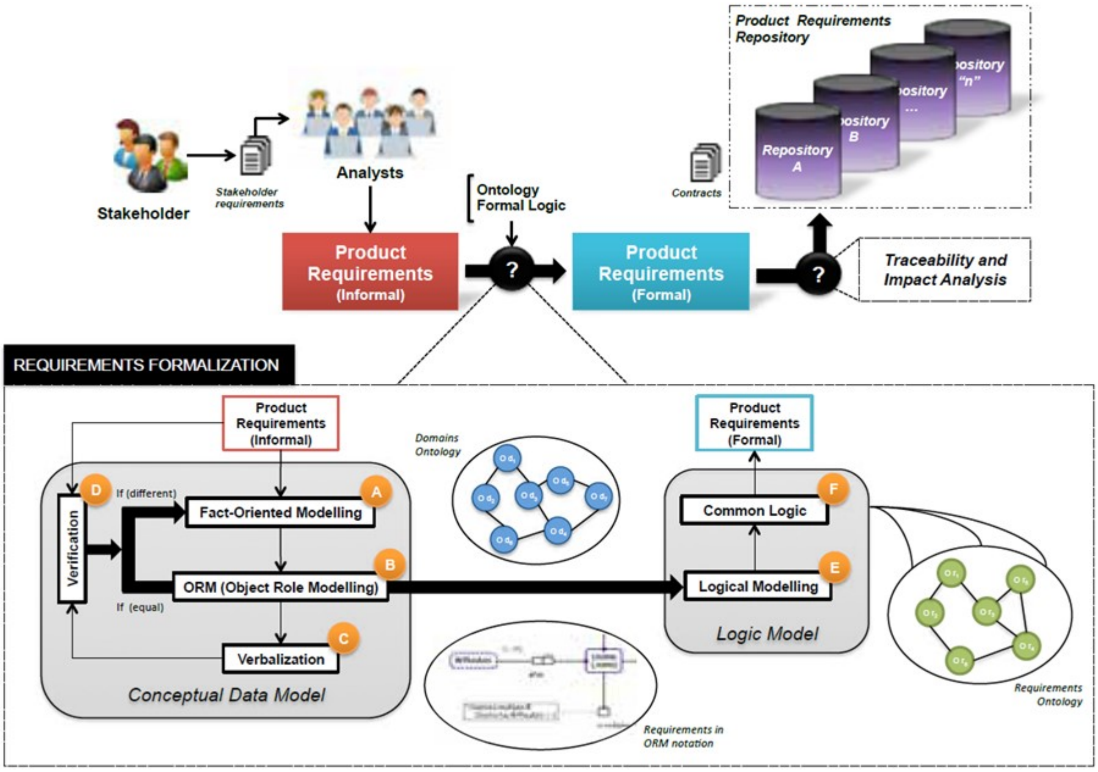


Figure 48 Product requirements formalization method (Szejka et al., 2014)

The method proposed by Szejka et al. (2014) is structured in two parts:
a) *Conceptual Data Model of Product Requirements*: consists in extract the information implicit in each requirement, identifying the main concepts of each sentence and dependence relations.

b) *Requirements Logic Model*: this part of the method transforms (translate, convert and share) the graphical model requirements to logical models.

Sometimes the stakeholder requirements are complex or, according to the analysis and design team experience, there is something missing and necessary for the system design; that is the reason of performing the next activity: requirement elaboration through decomposition and/or derivation. At the end of this activity, the stakeholder needs are already being translated into stakeholder requirements.

Activity: Requirement elaboration through Decomposition and through Derivation

In literature, diverse authors suggest performing the current activity at different moments of the process. In the QFD frame, requirement elaboration (decomposition and/or derivation) may be done: a) right **after** conducting the *gemba*, through the application of the Customer Voice Table when analyzing the verbatims, observations, and data, just before establishing the whole structure of stakeholder needs (number 4 in Figure 45) (Mazur, 2012; Bylund, Wolf, & Mazur, 2009); or b) **once** stakeholder needs are elicited, **prior** and to allow its prioritization (Faisandier, 2012; SEBoK, 2016; Ryan & Faulconbridge, 2016).

According to Faisandier (2012), SEBoK (2016) and Ryan & Faulconbridge (2016), the functional hierarchy of needs is achieved by performing two activities:

1. **Elicitation:** stakeholders literally express these elements through various techniques or methods.
2. **Elaboration:** This activity requires an analysis of the work team, as the elements are not expressed by stakeholders. It consists of two stages:
    a. **Decomposition:** consists of "breaking" a requirement of a certain level in another or other lower level requirements, are not explicitly requested.
    b. **Referral:** the work team must infer the requirement as it is not explicitly requested by stakeholders, but is necessary for the system design.

Activity: Divide, classify, and organize stakeholders requirements by type

According to SEBoK (2016) *type of requirement* refers to the *nature* of the requirement itself.

Similarly to the previous activity, this one may be performed at different moments of the process: a) **when** eliciting stakeholder needs: in this case, if requirement classifications are used at the moment of elicitation the stakeholder needs are already grouped by type (Sommerville, Sawyer & Viller, 1998; Hickey & Davis, 2003; Ryan et al., 2015; Ryan & Faulconbridge, 2016); or b) divide, classify, and organize stakeholders requirements by type **after** their elicitation; Faisandier (2012) propose to elicit needs, transform them into requirements, and after classify them to ensure that the analysis and design team have considered all kind of stakeholder requirements; the QFD frame, suggest to apply the Affinity Diagram (number 5 in Figure 45) for grouping requirements according to their affinity (Bylund, Wolf, & Mazur, 2009; Mazur, 2012).

In Chapter 2, it is described the state of the art of requirement classifications. Specifically, focusing on stakeholder requirements, Faisandier (2012) proposes a stakeholder requirement type classification. Figure 49 illustrates the different stakeholder requirement types:

1. Operational modes and operational scenarios
2. Expected services
3. Expected effectiveness or performance
4. Interfaces
5. Operational conditions
6. Constraints
7. Validation requirements

In some cases, it exists sub-types, and/or sub-sub-types, for instance for interfaces can be distinguished functional or physical interfaces.

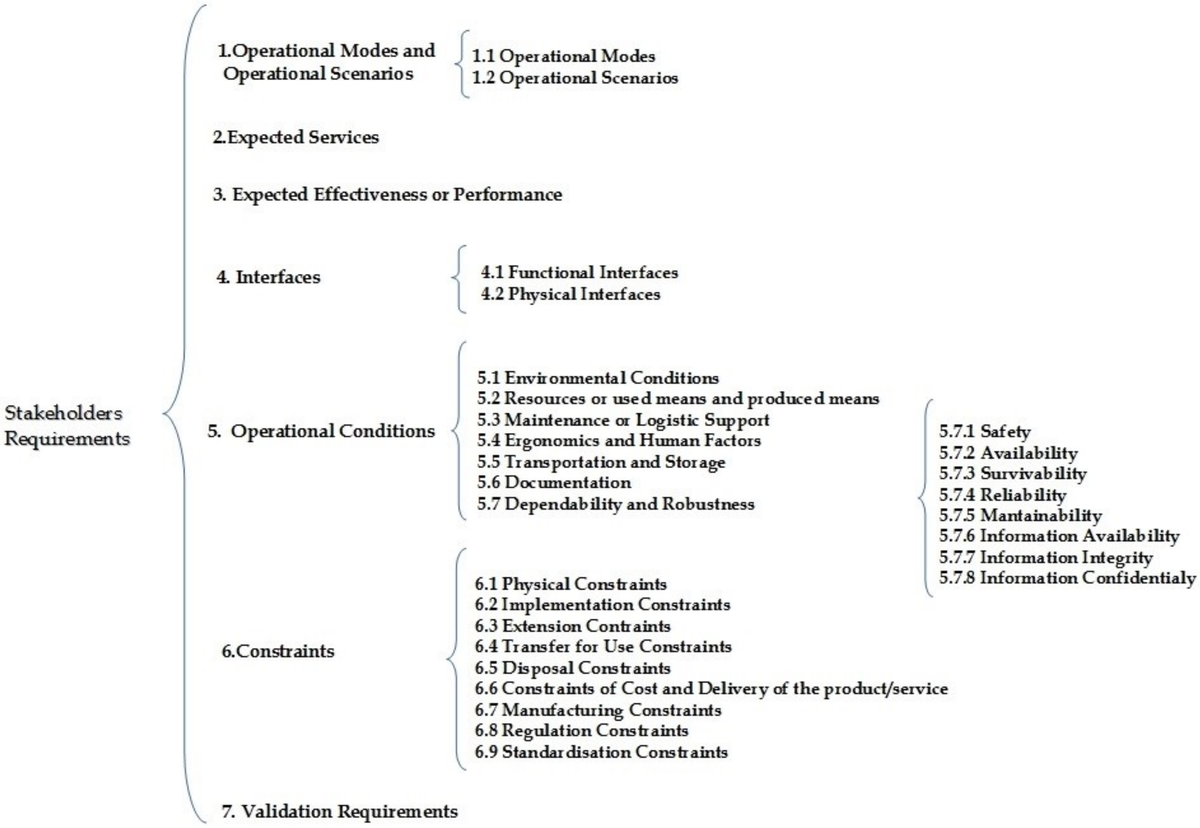


Figure 49 Stakeholder Requirement Classification according to Faisandier (2012)

Activity: Identify potential risks (or threats and hazards) that could be generated by the stakeholder requirements

Risk is one of the fifteen core information pointed by Sadig & Sahraoui (2017). A *risk* is an event that has a certain probability of occurrence (likelihood) and if it occurs, there will be gravity consequences (criticality) on the system mission or another characteristic; a risk combines vulnerability a danger, or threat (Faisandier, 2012).

One of the essential tasks of SE is dealing with risk; it requires a broad knowledge of the complete system and its critical elements, in order to decide how to balance risk and how the incurred risks should be reduced (Kossiakoff et al., 2011).

Costello & Ferguson (2010) states that risks can be identified using a number of processes. According to this author, the more common approaches for risk identification are presented in the following Table 34.

Table 34 Risk Identification Methods (Costello & Ferguson, 2010)

Formal	Informal
System safety assessments – fault tree analysis, hazard analysis, failure modes, and effects analysis	Brainstorming
Quantitative risk assessments	Test and verification
System and software engineering	Pause and learn sessions
Program planning and control – cost and schedule risk analysis	Experience – previous analysis, lessons learned, historical data
Models and simulations	

Bylund, Wolf, & Mazur (2009) recommend that a multitude of solution alternatives at an early stages should be followed by rigorous selection, in order to reduce the risk of running off with the first idea and pushing into a dead end; also suggest to do an early recognition of possible difficult challenges regarding manufacturing to reduce the risk of bad surprises late in the process.

According to De Weck, Eckert, & Clarkson (2007), *uncertainty* is a risk and an opportunity. These authors propose a checklist (Table 35) to manage uncertainty when designing a system; as a consequence, the analysis and design team can anticipate possible future changes in system design. This checklist includes the sources of uncertainty, their degree of resolvability, if the uncertainty is represented by a continuous or discrete variable and the possible modeling approach of the uncertainty variable. "*However, it should be clear that these methods can only help design for the known uncertainties, but the unknown unknowns will still occur and may either harmfully or beneficially impact the system or product… for exogenous uncertainties, these have to be projected into the system architecture and design embodiment to identify hardware and or software components that are most likely to be changed in the future*" (De Weck, Eckert, & Clarkson, 2007).

Table 35 Uncertainty Checklist in System Design (De Weck, Eckert, & Clarkson, 2007)

Sources of Uncertainty	Resolvability	Discreteness	Modelling Approach
Where does uncertainty come from that could affect future success of my system/product?	Can the uncertainty be resolved by simply delaying decisions and waiting until time *x*?	Can the uncertainty be represented as a random variable or as a discrete future scenario?	What modelling approach can be used to quantitatively capture the uncertainty?
Endogenous Product Context Corporate Context Exogenous Use Context Market Context Political/Cultural Context	Resolvable Irresolvable	Continuous Variable Discrete Scenario Probability-weighted Scenario	Geometric Brownian Motion Lattice Model Scenario List

Particularly, technical risks are present when the system cannot satisfy the requirements any longer, and the cause is the requirement itself and/or the solution (Faisandier, 2012).

According to SEBoK (2016), a technical risk analysis should be done to avoid risks; this analysis is based on three factors:

a. Analysis of potential threats or undesired events and their probability of occurrence (*likelihood*).
b. Analysis of the criticality of the *consequences* of these threats or undesired events and their classification on a scale of gravity.
c. Mitigation to reduce the probabilities of threats and/or the levels of harmful effect to acceptable values (*low, medium, or high*).

Nevertheless, every time a risk is identified, it should be treated as follows:

1) Transfer the risk: "*develop a strategy to place the risk with the party most able to do something about it*". If it is impossible to transfer the risk, then:
2) Assume the risk: *"accept consequences of the risk, with frequent monitoring to determine if the risk actually occurs and that the impact is as predicted (and is tolerable) if it does."*
3) Control or Mitigate the risk: *"develop a strategy to lower the risk by reducing its likelihood, consequence, or both components."*
4) Watch the risk: *"monitor and periodically re-evaluate the risk for change"*.

Risk management may be defined as *"the methodology that is employed to identify and minimize risk in the system"* (Kossiakoff et al., 2011). According to Kossiakoff et al. (2011), risk management may be studied as *risk assessment* (that includes *risk planning* and *prioritization*) and *risk mitigation* (which includes *risk handling* and *monitoring*).

Risk assessment is used to eliminate alternative concepts taking into account the likelihood and criticality (risk components) of every alternative. According to Costello & Ferguson (2010), risks can be prioritized by determining its *Priority Score*; this value may be assigned through the *Risk Matrix* (see Figure 50), a *risk cube* in five dimensions that shows the level of incurred risk: low (green), medium (yellow) or high (red).

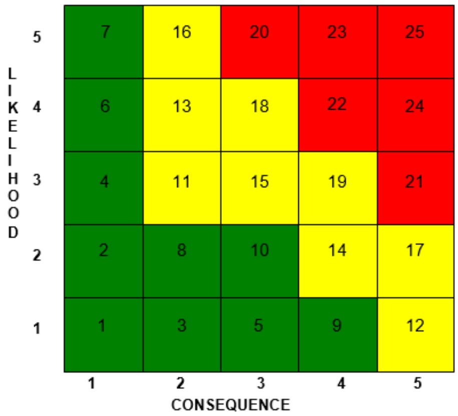


Figure 50 Risk Matrix (Costello & Ferguson, 2010)

The Risk Matrix is based on the risk's likelihood and consequences. Table 36 shows the likelihood criteria: the score, its description, and its probability of occurrence.

Table 36 Risk likelihood criteria (Costello & Ferguson, 2010)

	Likelihood	
Score	Likelihood of Occurrence (p)	
5	Near certainty	p > 80%
4	Highly Likely	60% < p ≤ 80%
3	Likely	40% < p ≤ 60%
2	Low likelihood	20% < p ≤ 40%
1	Not likely	p ≤ 20%

In Table 37 is shown the risk criticality criteria, analyzed and scored by separate consequence category.

Table 37 Risk criticality criteria (Costello & Ferguson, 2010)

Score	CONSEQUENCE				
	1	2	3	4	5
Performance	The minimal consequence to objectives/goals	The minor consequence to objectives/goals	Unable to achieve a particular objective/goal, but remaining objective goals represent better than minimum success or outcome	Unable to achieve multiple objectives/goals but minimum success can still be achieved or claimed	Unable to achieve objectives/goals such that minimum success cannot be achieved or claimed
Safety Human	Discomfort or nuisance	First aid event per OSHA criteria	No lost time injury or illness per OSHA criteria	Lost time injury or illness per OSHA criteria	Loss of life
Asset	Minimal consequence: asset has no sign of physical damage	Minor consequence: asset has cosmetic damage and is repairable	Minor consequence: an asset is damaged but repairable	Major consequence asset is substantially damaged but repairable	Destroyed: an asset is compromised, and un-repairable: a total loss
Schedule	Minimal consequence	The critical path is not slipped; the total slack of slipped tasks will not impact the critical path in less than 10 days	The critical path is not slipped; total slack of slipped tasks is within 10 days of impacting the critical path	Critical path slips	Critical path slips and one or more critical milestones or events cannot be met
Cost	Minimal consequence	Minor cost consequence. Cost variance ≤ 5% of total approved FY baseline	Cost-consequence. Cost variance >5% but ≤ of total approved FY baseline	Cost-consequence. Cost variance >10% but <15% of total approved FY baseline	Major cost consequence. Cost-consequence. Cost variance >15% of total approved FY baseline

Costello & Ferguson (2010) suggests that "*risk identification shall begin as early as possible and continue throughout the project lifecycle*".

Activity: Verify quality stakeholders requirement (against good requirement characteristics)

Considering that stakeholder requirements may be incomplete, inconsistent and/or poorly expressed, the interdisciplinary team should check, before continuing, to ensure that stakeholder requirements are complete, consistent, include expected functions, constraints, and operational scenarios. This step is known as Verification.

Faisandier (2012) suggests verifying the requirements through checking the quality of each requirement statement by comparing it against the quality characteristics of a *good* individual requirement and the quality characteristics of a set of requirements (see Chapter 2).

In addition, Faisandier (2012) recommends put special attention to:

- ***Maturity:*** Does the expression of stakeholder requirements approach to the stakeholder expectations?

- ***Exhaustiveness:*** Were all stakeholders identified and interviewed? Were all stakeholder requirements expressed and written?
- ***Accuracy:*** Did stakeholders express their expectations with precision?
- ***Feasibility:*** Is the feasibility of stakeholder requirements evaluated through identified operational concepts?
- ***Consistency:*** Is there any conflict among stakeholder requirements?

Once stakeholder needs are captured, it should be verified through a feasibility study (Haramis, 1992; Currie, Seaton, & Wesley, 2009), if the set of stakeholder requirements are feasible and consistent among them, in other words, if they fit one with another, if they are compatibles, not opposites. Bylund, Wolf, & Mazur (2009) point out that knowing the most critical characteristics in the product (system) to develop helps to eliminate contradictory development.

Schillo, Isabelle, & Shakiba (2017) state that "*QFD provides a rigorous analysis of the needs of stakeholders*". This analysis is done by pairs of stakeholder needs to determine their co-relationship; Hauser & Clausing (1988) proposes the following scale: strong positive (✓), medium positive (✓), medium negative (×), or strong negative (×).

Note that in the QFD frame the authors only mention *customer needs*, they do not make difference between *customer needs* (stakeholder needs) and *customer requirements* (stakeholder requirements).

In Figure 51 it can be observed as an example a small part of the house of quality, proposed by Houser & Clausing (1988) and currently part of the Blitz QFD method; there are listed the *customer needs*, and their co-relationships by pair comparisons are expressed by symbols.

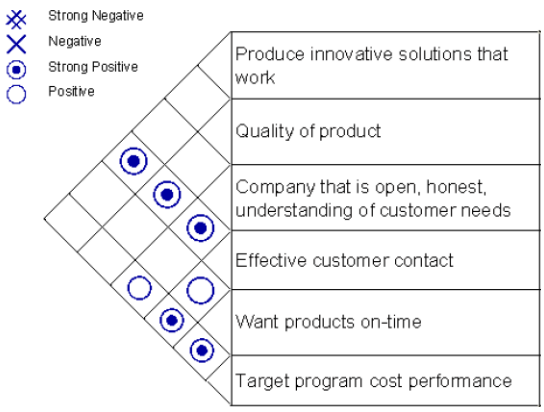


Figure 51 Example of the co-relationship matrix of 'customer needs'

In case that any pair, or more, are identified to be incompatible (not feasible, nor consistent, or both), the stakeholders are encouraged to negotiate until consensus (Van Lamsweerde, 2000; Faisandier, 2012; Bradreau & Boulanger, 2014; SEBoK, 2016; Hoh In, Boehm, Rodger, & Deutsch, 2001; Savage et al., 2011; and Narendhar & Anuradha; 2017). "*An unresolved conflict over the requirements will threaten the use of the system, its acceptance and therefore, the success of the project*" (Badreau & Boulanger, 2014).

According to Boehm (2001) to identify and resolve this kind of conflicts is difficult, and the difficulty comes from:

- Stakeholder interests and priorities are different

- Dependencies among quality attributes are complex
- Conflict resolutions exist within an exponentially increasing solution space

Pohl & Rupp (cited in Badreau & Boulanger, 2014) identify four activities to negotiate the requirements:

1) **Identify the conflict:** the conflicts can arise during the entire system life cycle
2) **Analyze the conflict:** to be effective it is necessary to understand the nature or type of the conflict; Badreau & Boulanger (2014) describe four types de conflict among the requirements, see Table 38.

Table 38 Types of conflicts among requirements (adapted from Badreau & Boulanger, 2014)

Type of conflict	Description
Conflict on the subject	One requirement is understood in different ways by the stakeholders
Conflict of interest	The stakeholder objectives are divergent
Value conflict	The stakeholders value the same criterion differently, divergence from the personal ideal
Structural conflict	The level of influence of conflicting stakeholders is unequal

3) **Resolve the conflict:** there are several techniques to solve requirement conflicts (see Table 39); nevertheless, it is very important to pay attention to which technique is going to be applied because it will impact the team motivation (Badreau & Boulanger, 2014).

Table 39 Requirements conflict resolution techniques (Badreau & Boulanger, 2014)

Technique	Description
Agreement	Negotiation of one agreement
Compromise	Combination of alternative solutions
Vote	Voting among a list of alternative solutions
Variant formation	Variants of the system will be able to satisfy conflicting requirements
The leader always right	Hierarchical decision
Consider all of them done	Systematic consideration of factors that influence the conflict
Strengths and weaknesses	Ponderation of the factors that influence the conflict
Decision Matrix	Evaluation matrix according to a list of criteria

One model of the *Agreement technique* is the *Win-Win negotiation model*; it provides a general framework to identify and solve conflicts through drafting and negotiating artifacts: *win conditions, issues, options,* and *agreements* (Hoh In, Boehm, Rodger, & Deutsch, 2001).

The Win-Win negotiation model is based on Theory W: *"make everyone a winner"* (Boehm, 2001 Applying), or "*make winners all success-critical stakeholders*" (Narendhar & Anuradha, 2017). It involves the stakeholders who identify their *win conditions* and reconcile conflicts among them.

According to Narendhar & Anuradha (2017), the key activities of Win-Win negotiation model are:

a) To identify the success-critical stakeholders (who will negotiate and make decisions)
b) To elicit the primary win conditions
c) To negotiate the mutually satisfactory win-win situation packages
d) To monitor and control a value-based win-win equilibrium

The resulting negotiating artifacts are illustrated in Figure 52. The stakeholders enter the *win conditions* in the *win condition schema;* if there is a conflict among *win conditions*, the stakeholders develop an *issue schema* that summarizes the conflict and the involved *win conditions*; the stakeholders prepare a candidate *option schema* addressing every issue, the stakeholders evaluate the options to converge on a mutually satisfactory option; finally, the adoption of this option is formally expressed in the *agreement schema.* Note: the taxonomy organizes the Win-Win artifacts (Narendhar & Anuradha, 2017).

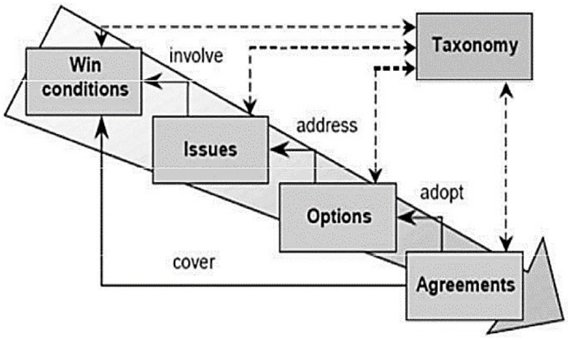


Figure 52 WinWin Artifact Relationships and Taxonomy (Narendhar & Anuradha, 2017)

When applying the Win-Win model in a real case, Hoh In, Boehm, Rodger, & Deutsch (2001) had some interesting findings:

a) The stakeholders accept *satisfactory* rather than *optimal* solutions
b) Stakeholder interests can not necessarily be anticipated
c) Stakeholders like *users* and *customers* are proactive in stating *win conditions*; on the other hand, *developers* are more active in working to find a solution
d) Knowledge-based semi-automated aids, like QARCC (Quality Attribute Risk and Conflict Consultant) and S-COST (Software Cost Option Strategy Tool), provide improvements in conflict identification and resolution

According to Narendhar & Anuradha (2017), there are several methods to negotiate requirement conflicts, and practitioners may choose *the best* depending on the conflict type. These authors highlight that Theory W (Win-Win approach) is very popular due to "*resource conflicts and feasibility conflicts still are getting almost no attention at all*".

4) **Document the conflict resolution:** to avoid re-work when the team tries to solve the same conflict later; Badreau & Boulanger (2014) suggest documenting:
   - **a.** The origin of the conflict
   - **b.** The involved stakeholders
   - **c.** The different stakeholder opinions
   - **d.** The different means used to resolve the conflict
   - **e.** The eventual alternatives
   - **f.** The decision made
   - **g.** And the reasons to make the decision

Once stakeholder requirements are verified, the analysis and design team is sure that they are going to *build right* the system. But it is imperative to validate if they are going to build the *right system*, then, the stakeholder requirements must be validated.

Activity: Validate stakeholders requirements

Stakeholders must validate the content value, relevance, and significance of the requirements defined at this stage. Faisandier (2012) suggests validating in particular:

- *Understanding:* each stakeholder owner of the requirement must validate that it is understandable, and confirm that the resolution of the conflict with other requirements does not compromise its intentions.
- *Relevance*: the expression of requirements allows to define the importance of the solution?
- *Justification:* why do these needs and expectations exist? What risk or cause could make these expectations disappear?

References Mazur (2012) and Bylund, Wolf, & Mazur (2009) suggest, as part of the QFD framework, the definition of customer (stakeholders) metrics. This values will be useful at the time of validating stakeholder requirements, they document them in the CVT.

At the time that stakeholder requirements are verified and validated, it is time to move on and advance.

Activity: Synthesize, record, and manage the stakeholder requirements and potential associated risks

Faisandier (2012) suggest synthesizing similar requirements in order to reduce the quantity of them. This synthesis refers only to those requirements that are repeated.

It is well known that requirements, even needs, change through time (Ryan & Faulconbridge, 2016), and these changes affect not only the involved requirement, but they also affect the relationships among requirements, and the system synergy (see Chapter 2). That is the importance of keeping requirement information available through all system life cycle (Faisandier, 2012; SEBoK, 2016; Ryan et al., 2015; Ryan & Faulconbridge, 2016). To document the requirements will allow their communication, inspection, negotiation, and evolution (Van Lamsweerde, 2000).

Currently, there are automatized tools in the market to draft, record, and manage requirements; but the need of a better tool still remains (Faisandier, 2014; Cuiller, 2015; Ryan & Faulconbridge, 2016; and Badreau & Boulanger, 2014).

Mazur (2012) and Bylund, Wolf, & Mazur (2009) record the stakeholder requirements in different artifacts mentioned before, like GVT, CVT, CPM, AD, and HD; these authors claim that if QFD approach is used, the software may be needed depending on the project and goals. Mazur, Terniko, & Ziltner (2017) assures that is more important to know *how to do the job* that developing a very sophisticated tool (software) to support the process; the author suggests the use of Excel© and Word©.

Back to the potential associated risks to the stakeholder requirements, once they are identified, additionally they need to be managed.

Kehl et al. (2014) proposes a risk mitigation plan worksheet for performing risk management. It can be seen in Figure 53. The information includes the risk title, risk statement, context statement, closure criteria, the likelihood and consequence graph, the date of discovery of the risk (sunset), the latest date that the risk was an issue (sunset), the impact horizon of the risk (near, mid, or long future), the consequence score, its rationale, the likelihood, and its rationale.

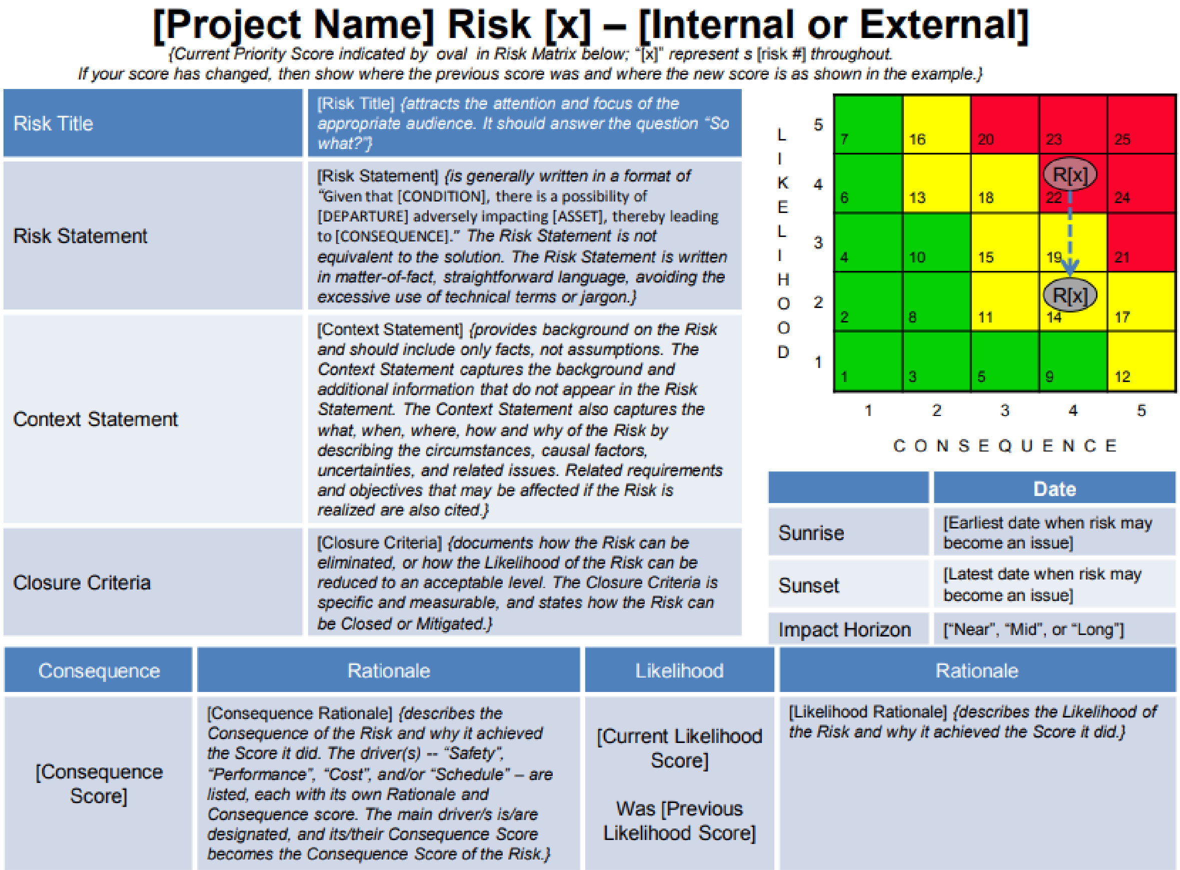


Figure 53 Sample risk plan worksheet (Kehl et al., 2014)

Kossiakoff et al. (2011) state as common methods for *risk mitigation*: technical and management reviews, oversight of designated component engineering, special analysis and testing, rapid prototyping, relief of excessive requirements, and fallback alternatives. The author highlights the importance of formal risk management; they suggest that for every significant risk there should be a plan that helps the minimization of potential impacts in the project.

According to Kossiakoff et al. (2011), "*reducing program risk is a continual process throughout the system life cycle*" (see Figure 54); in this figure it can be seen how the risk decreases while the relative development effort increases; it is, at the beginning of the life cycle the incertitude and risk are high because of the unforeseen adverse events, but after analysis, experiment, test, or change, this uncertainty is reduced.

As Kossiakoff et al. (2011) express, Figure 54 reveals certain key principles that demonstrate the importance of investing resources in the initial stages of the system life cycle in order to reduce risk:

- To maintain program support, the risk of failure must be reduced to maintain the financial risk at reasonable levels.
- The initial stages in the program allow major reduction of risk: decision on system requirements and system concept.
- Concept exploration and advanced development generate the greatest risk reduction
- When the system is ready for production, the level of risk is almost null.

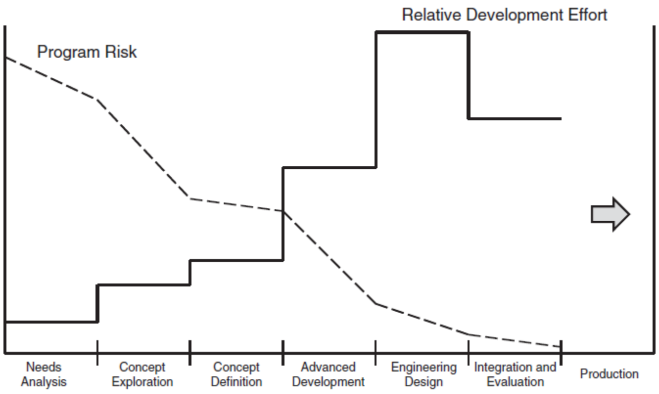
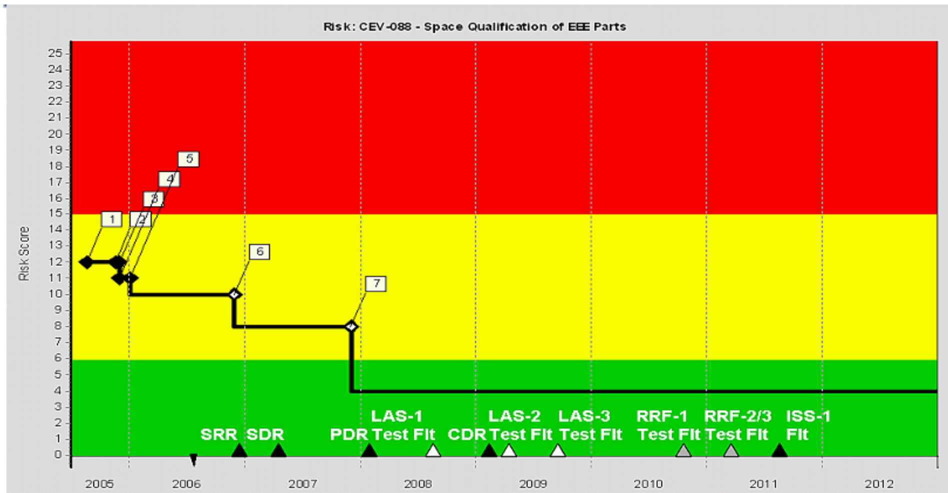


Figure 54 Variation of program risk and effort throughout system development (Kossiakoff et al., 2011)

Costello & Ferguson (2010), Kehl et al. (2014), and Kossiakoff et al. (2011) proposes the application of the Risk Waterfall Chart (see Figure 55) to follow through time the evolution of every risk; this chart helps the visualization of the risk behavior, it shows the key mitigation action planned, its number or identifier, the success criteria, the cost of the action, the date the action started and the date of action completion.



Key Mitigation Action Planned	Success Criteria	IMS#	Cost	Start	Compl.
01 Initial Risk Assessment	Reviewed and approved by the Risk Management Board.	1737	*	8/15/05	8/18/05
02 Initial Radiation Analysis	SEU and TD Levels Identified.	1738	*	10/3/05	11/17/05
03 Initial MIMU Assessment Capability	Established ability of existing parts and components to meet levels.	1739	*	10/10/05	11/30/05
04 PPC-750 & PPC-74XX Evaluation	Established ability of other PPC processors to perform in environment.	1740	*	10/10/05	11/30/05
05 Avionics Subsystem Specification	Radiation requirements established.	1741	*	11/21/05	1/2/06
06 Performance Verification Plan	Plan and schedule established to conduct testing.	1742	*	5/1/06	11/30/06
07 Radiation Testing & Evaluation Complete	Successful test completion.	1743	$251K	5/1/07	12/3/07

Figure 55 Example of Risk Waterfall Chart (Kehl et al., 2014)

3.3 Discussion

After reviewing literature, it can be said that authors like Faisandier (2012), Badreau & Boulanger (2014), ISO 15288 (2015), SEBoK (2016), Ryan et al. (2015), Ryan & Faulconbridge

(2016), and Aasheim & Zhao (2017) agree almost in all the activities to perform in order to translate stakeholder needs into stakeholder requirements.

Nevertheless, there is a large amount of information regarding *how* to conduct these activities, and in some cases, *when* is the most convenient moment to do it. It remains the opportunity to design a lean process that includes the activities that add value while waste is reduced and, at the same time, assures the quality of the product.

3.4 Conclusion

Chapter 3 presented the state of the art of how to translate stakeholder need into stakeholder requirements.

The different activities to transform stakeholder needs into stakeholder requirements were identified, and different propositions of several researchers were presented. The activities are: identify the stakeholders across the life cycle; elicit, capture, and consolidate the stakeholder needs, expectations, objectives, and constraints; prioritize the retained stakeholder needs; transform the prioritized and retained stakeholder needs into stakeholder requirements; stakeholder requirement elaboration through decomposition and through derivation; divide, classify, and organize stakeholders requirements by type; identify potential risks, threats and hazards that could be generated by the stakeholder requirements; verify the quality of stakeholders requirements; validate the stakeholder requirements; synthesize, record, and manage the stakeholder requirements and potential associated risks..

Chapter 4 will present the state of the art of how to translate stakeholder requirements into system requirements.

CHAPTER 4 Stakeholder Requirements to System Requirements

4.1 Introduction

This chapter presents the state of the art to translate stakeholder requirements into system requirements. It gives the different points of view of authors in Systems Engineering like Hull, Jackson, & Dick (2011), Kossiakoff et al. (2011), or SEBoK (2016) that propose a number of activities to carry out this process; besides, this chapter describes and illustrates how several authors perform these activities.

4.2 State of the art

Figure 56 illustrates the process of translating stakeholder requirements into system requirements, it is located in the Concept stage of the system life cycle.

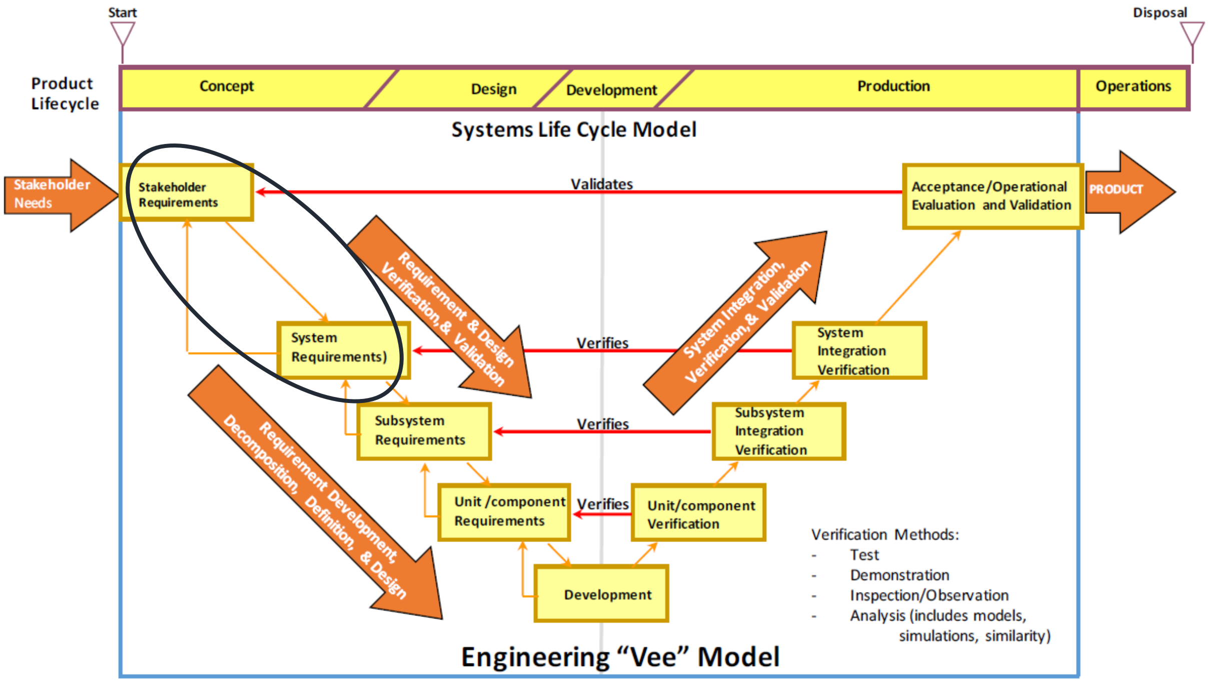


Figure 56 Localization of the process within the development cycle (adapted from Ryan & Wheatcraft, 2017)

According to several authors (SEBoK 2016; ISO 15288, 2015; Cuiller, 2015; Ryan & Faulconbridge, 2016), the aim of this process is to define a coherent set of quantifiable, feasible, achievable and verifiable technical requirements applicable to the system.

Chapter 3 explained how stakeholder needs are translated into stakeholder requirements; in this chapter, Figure 57 illustrates how the *stakeholder requirements* (retained needs) are translated into *system requirements* (specified needs); this translation means that exist potentially feasible solutions. This step is essential to define, as precisely as possible, the technical and design characteristics for the future system.

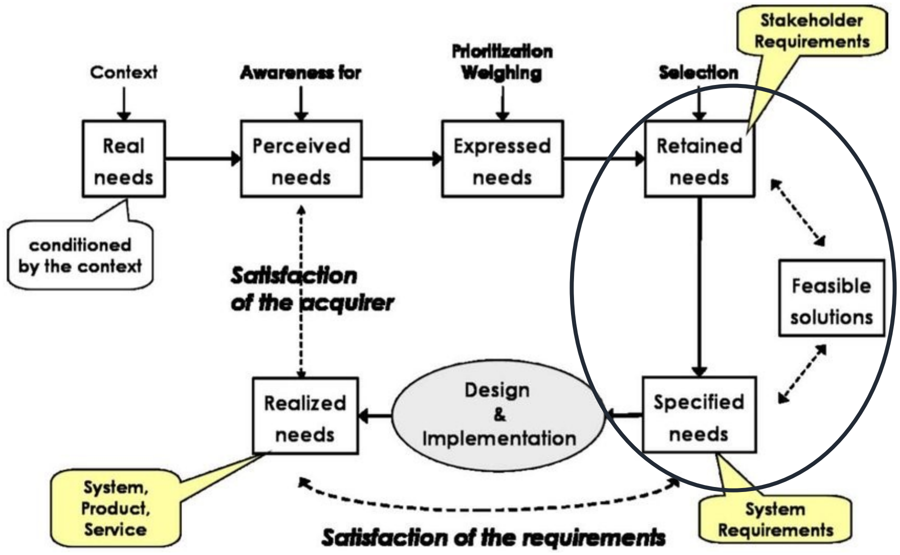


Figure 57 Cycle of needs and requirements (adapted from Faisandier, 2012, p.82)

Faisandier (2012) proposes a process for the definition of the system requirements, it is illustrated in Figure 58. The proposed activities of the process are carried out in sequence, but iterations are necessary until mature and consistent requirements are obtained. The inputs for system requirements definition process may include the set of consistent stakeholder requirements, information of stakeholder interviews, and feedback from verification, validation, and analysis processes. Once the input information is complete, the stakeholder requirements should be analyzed or completed if needed. Immediately, it is possible to define, consolidate, and organize the system requirements, to finally verify and validate them. Inevitably, system requirements will evolve through time, thus, the management of system requirements is necessary to update them. As can be seen, the generic output of this process will be the documented baseline of the set of consistent system requirements.

"*System modeling supports the analysis and design process by introducing a degree of formality into the way systems are defined*" (Hull, Jackson, & Dick, 2011). According to SEBoK (2016), there are several modeling techniques that can be applied to detail or refine system requirements, for example: state-charts models, scenarios modeling, simulation, prototyping, Quality Function Deployment (QFD), systems modeling language (SysML), sequence diagrams, activity diagrams, use cases, state machine diagrams, requirements diagrams, and/or Functional Flow Block Diagram (FFBD) for operational scenarios. Hull, Jackson, & Dick (2011) explain that the used modeling techniques vary from one industry to another; for example, aircraft industry employs aerodynamic models, three-dimensional spatial models, weight distribution models, and/or flight simulators; rail industry uses timetable simulations, and/or safety, reliability and maintainability models; and car industry apply styling models, dashboard models, and/or aerodynamic models.

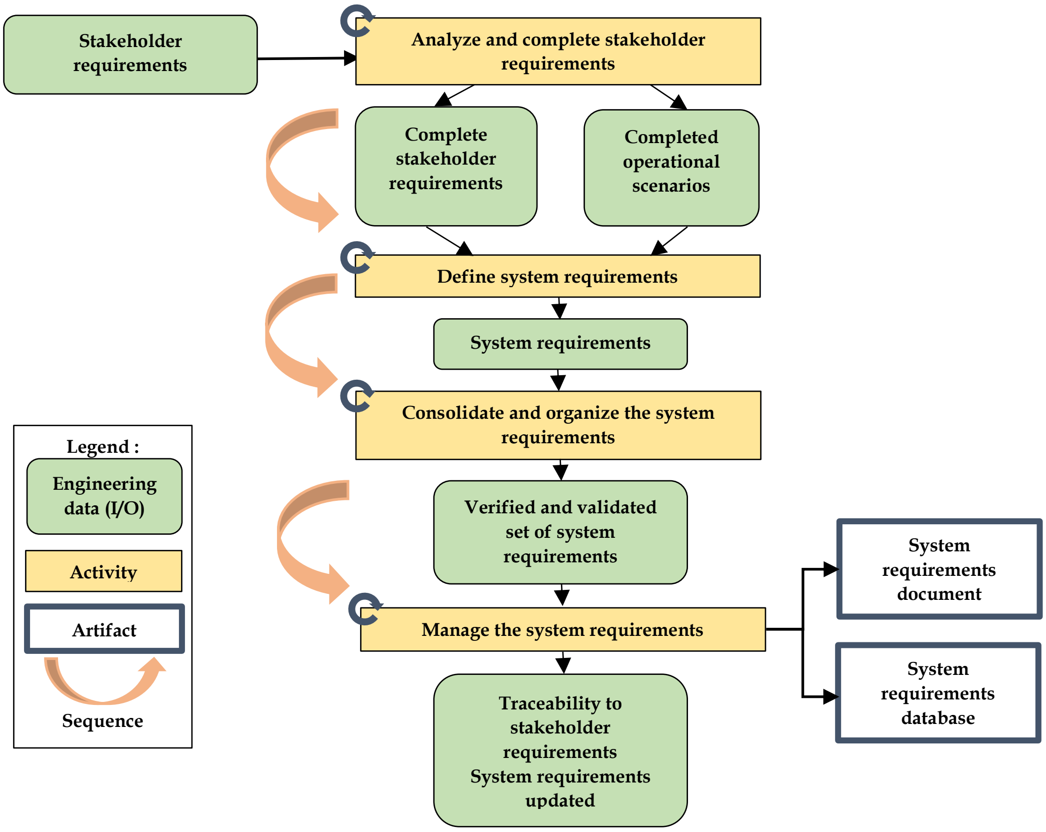


Figure 58 Activities of the System Requirements Definition Process (Faisandier, 2012, p.139)

This research work is conducted under Systems Engineering philosophy, that is the reason why in Table 40 below are shown the activities proposed by Systems Engineering's authors to translate stakeholder requirements into system requirements. These activities were identified through the analysis of findings in the literature and do not follow a strict order, this is because authors differ in their point of views.

Table 40 Activities to translate stakeholder requirements into system requirements by author

Activities	Hull, Jackson, & Dick (2011)	Faisandier (2012, 2014)	Badreau & Boulanger (2014)	ISO 15288 (2015)	SEBoK (2016)	Ryan et al. (2015) Ryan and Faulconbridge (2016)
Analyze stakeholder requirements to verify the completeness	•	•	•	•	•	•
Transform stakeholder requirements into system requirements, and define their rationales	•	•	•	•	•	•
Incorporate the derived and decomposed requirements (coming from architecture and design) into the system requirements baseline	•	•	•	•	•	•
Classify the system requirements according to their type	•	•	•	•	•	•
Establish the upward traceability with the stakeholder needs and requirements	•	•	•	•	•	•
Establish bi-directional traceability between requirements at adjacent levels of the system hierarchy					•	
Identify potential risks (or threats and hazards) that could be generated by the system requirements	•	•	•	•	•	•

Verify the quality and completeness of each system requirement and the consistency of the set of the requirements	•	•	•	•	•	•
Validate the content and relevance of each system requirement against the set of stakeholder requirement	•	•	•	•	•	•
Synthesize, record, and manage the system requirements and potential associated risks	•	•	•	•	•	•

Numerous authors such as Altshuller (2000), Hull, Jackson & Dick (2011), Kossiakoff et al. (2011), or (Mokammel et al., 2018) to name a few, have tried to improve the way these activities are performed. In the following subsections, the reader will find several author's contributions according to each activity.

Activity: Analyze stakeholder requirements to verify the completeness

Badreau & Boulanger (2014) describe this analysis of stakeholder requirements through the following process:

a) Start studying the stakeholder requirements and accept them
b) Consider equally other sources like standards, architectural and project constraints
c) Produce the system (and sub-system) requirements as the functions the system shall perform, instead of *how* the system shall perform the functions.
d) Link the system (and sub-system) requirements to the stakeholder requirements and other sources, justifying this relation with stakeholder need satisfaction.
e) Model the system requirements with aid of activity, classes or sequence diagrams
f) Define the requirements derived from the verification strategy

Hull, Jackson, & Dick (2011) suggest reviewing the stakeholder requirements with rigor, thoroughness, and "*do not add more detail than is necessary*" to produce the system model that defines the requirements oriented to *what* should be done rather than *how*.

Faisandier (2012) recommends that the system engineers, or the supplier, review the exhaustiveness and consistency of the provided stakeholder requirements before starting any further activity. The author suggests performing the following activities iteratively:

- Complete the list of stakeholders and identify/define their needs, expectations and/or constraints, for example enterprise context, procedures, environmental conditions, program constraints, available resources and available technologies, interfaces with existing systems, etc;
- Analyze and complete the operational scenarios;
- If possible, define other life-cycle stages scenarios;
- Consolidate and organize the completed stakeholder requirements.

At this step of the analysis, it is important to deduce the functional and physical interfaces of the system. The functional interface includes the input-output flows of material, energy and/or information between the system and the different elements of its context: a) the existent systems which the new system will interact, b) the stakeholders who are intended or interact with the system, c) the risk or threats against the system, and d) the adverse effects that must be prevented (Hull, Jackson, & Dick, 2011). The physical interface considers: the necessary physical connections to carry the exchanged flows of material, energy and/or information (Faisandier, 2012).

As said before, diagrams can be useful tools to express the functional and physical interfaces. One example is shown in Figure 59, where the system of interest (SoI) is a Terrestrial Fire Intervention Vehicle named FITVEE; it interacts with many entities providing or receiving services, materials, or information; for example, a) the FITVEE exchanges information and request reinforcement with the coordination and communication center (COCOCE), the COCOCE sends alert to the FIFVEE; b) the water distribution network provides water to the FITVEE, etc.

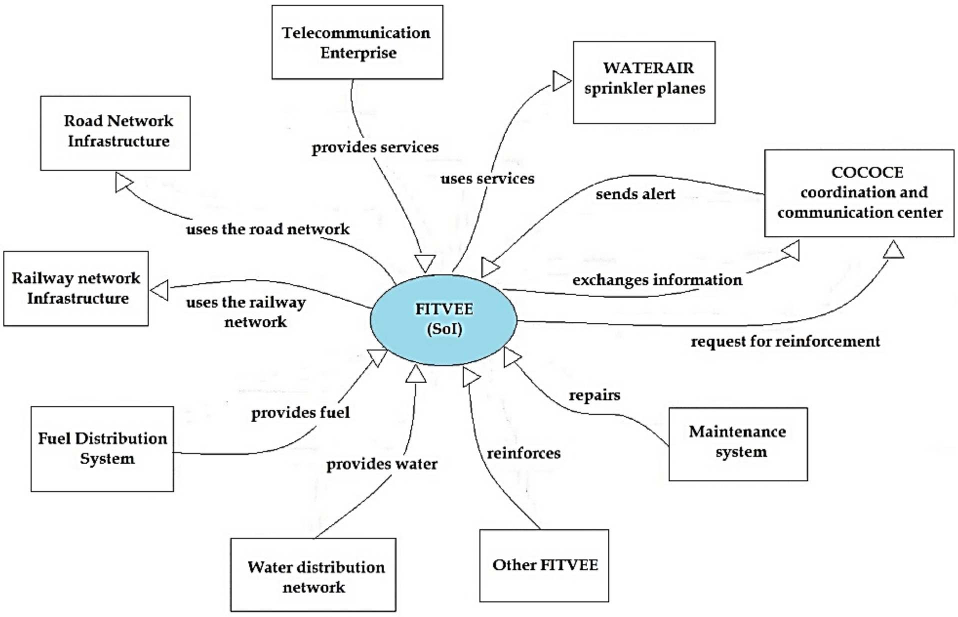


Figure 59 Example of Context Relationships Diagram (Faisandier, 2012)

Activity: Transform stakeholder requirements into system requirements, and define their rationales

The analysis and design team must translate the *stakeholder requirements* into *systems requirements*, consisting of expressing them in a technical language.

Chapter 2, section 2.3 described the state of the art of how to express requirements and the quality characteristics they should possess. Nevertheless, it can be added some additional recommendations.

Common mistakes when writing System Requirements

Faisandier (2012) and SEBoK (2016) agree on which are the most serious errors during the definition of system requirements:

- Insufficient analysis of stakeholders requirements
- Insufficient analysis of operational modes and scenarios
- Incomplete set of system requirements
- Lack of verification method
- Missing traceability

Proven practices when writing System Requirements

- ✓ Check that stakeholder requirements are complete before the definition of the system requirements (Faisandier, 2012; Badreau & Boulanger, 2014; Ryan et al., 2015; Ryan and Faulconbridge, 2016; SEBoK, 2016)
- ✓ Pay particular attention to the system's "inputs" and "outputs" (Cuiller, 2015)
- ✓ Focus on *key* requirements (Hull, Jackson & Dick, 2011; Cuiller, 2015)
- ✓ Involve the stakeholders as soon as possible while developing system requirements (Faisandier, 2012; Oehmen, 2012; SEBoK, 2016)
- ✓ Use short sentences (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Use of modeling techniques (Hull, Jackson & Dick, 2011; Faisandier, 2012; SEBoK, 2016; Ryan & Faulconbridge, 2016)
- ✓ Paraphrasing stakeholder requirements (Cuiller, 2015)
- ✓ Rethinking stakeholder requirements in technical language (Cuiller, 2015)
- ✓ One concept should be determined by one or several terms and should be used everywhere as this term or this set of terms (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Define a glossary for words having a general meaning different of the technical meaning (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Translate a single idea into a single requirement (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Begin the sentences with: "*The system should*... or *The system shall*...." (Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015)
- ✓ Requirements should be written in simple sentences: subject + verb in present tense + complement (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Review that requirements meet the quality characteristics of "system requirements" and "set of system requirements". (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015; Mokammel et al., 2018)
- ✓ Justifying the requirements through a rationale. (Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015; Ryan et al., 2015; Ryan & Faulconbridge, 2016; SEBoK, 2016)
- ✓ Requirements should be SMART (Specific, Measurable, Achievable, Realistic, Testable) (Faisandier, 2012),
- ✓ Organize peer reviews with applicable experts (Faisandier, 2012; SEBoK, 2016)
- ✓ Use typical measures for requirements engineering for volatility, trends, verification progress, validation progress, and peer review defects (Faisandier, 2012; SEBoK, 2016)
- ✓ Consider using a requirements management tool to trace linkages between system requirements and to display their relationships (Faisandier, 2012; SEBoK, 2016)
- ✓ Ensure that all types of requirements are considered (Faisandier, 2012; Cuiller, 2015)

Issues about wording:

- ✓ Adjectives should be measurables and quantifiables to allow verification (Faisandier, 2012)
- ✓ Avoid sentences with double negation (Faisandier, 2012; Badreau & Boulanger, 2014),
- ✓ Avoid orienting the solution (Faisandier, 2012)
- ✓ Avoid subjective sentences (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Avoid indefinite, generic or meaningless terms (Faisandier, 2012)
- ✓ Avoid speculation (Hull, Jackson, & Dick, 2011)

- ✓ Avoid wishful thinking (example: 100% reliable) (Hull, Jackson, & Dick, 2011)
- ✓ Avoid vague adjectives as: some, any, different, various, several, fast, correct, good, serious, and critical (Hull, Jackson, & Dick, 2011; Faisandier, 2012)
- ✓ Avoid vague adverbs as: little, much, enough, less, more, after, front, often, sometimes, a long time, at once, quickly, well, almost, and correctly (Hull, Jackson, & Dick, 2011; Badreau & Boulanger, 2014)
- ✓ Consult the dictionary as frequently as possible (Faisandier, 2012)
- ✓ Avoid common drafting errors such as contamination, redundancy, implicit requirements, inaccuracies, contradiction, ambiguities or omissions (Hull, Jackson & Dick, 2011; Faisandier, 2012)

According to Hull, Jackson & Dick (2011), Faisandier (2012), and Cuiller (2015), it is practically impossible to write the set of complete and correct system requirements from the first attempt, it is usual to find three or four versions prior to the final version; they mention that the key to success is the correct administration of the evolution of the requirements. "*Expressing the requirement even if it is vaguely will be better than not writing it*" (Faisandier, 2012).

Several authors (Faisandier, 2012, 2014; Ryan & Faulconbridge, 2016; SEBoK, 2016 for example) converge in the suggestion of the definition of the *rationale* for every system requirement; it is, the answer of the question: *why* does this system requirement exist? The first justification for the existence of a system requirement should be the existence of the stakeholder requirement from which it was originated, as well as including the justification for its importance. Recording the rationale will avoid rework and loss of time to the analysis, design team, and stakeholders; they will not have to ask or been asked again for the reason of the existence of the requirement.

Activity: Incorporate the derived and decomposed requirements (coming from architecture and design) into the system requirements baseline

The objective of this activity is to complete the set of system requirements; it is carried out by the analysis and design team.

Various authors such as SEBoK (2016), Faisandier (2012) and Ryan & Faulconbridge (2016) propose the following classification according to the type of assignment of the system requirements (Table 41). This classification aims at helping the analyst or designer to know where the requirement comes from: stakeholder needs elicitation process (direct assignment), complex requirements decomposed (indirect assignment) a) by a simple transformation, or b) with aid of a modeling technique, and architecture/design decisions (derived requirements).

Table 41 Type of assignment of system requirements

Type of assignment	The requirement comes from…
Direct assignment	The requirement comes from stakeholder needs elicitation process (Ryan & Faulconbridge, 2016) without any transformation (Faisandier 2012). These requirements are assigned directly from the higher level to a lower level.
Indirect assignment (simply decomposed)	The requirement comes from more complex requirements which are decomposed (Ryan & Faulconbridge, 2016) with a simple transformation (Faisandier, 2012)
Indirect assignment (modeled and decomposed)	The requirement comes from more complex requirements which are decomposed (Ryan & Faulconbridge, 2016) with the aid of a complex (modeled)

	transformation (Faisandier, 2012). These requirements are distributed to several systems using an analysis or modeling technique.
Derived requirement (from design)	These requirements are a "must" to the system (Ryan & Faulconbridge, 2016) that come from architecture or design decisions, not from stakeholder requirements (Faisandier, 2012). These requirements are developed as a result of design team decisions, not the stakeholder community.

Activity: Classify the system requirements according to their type

As mentioned before, SEBoK (2016) states that the type of requirement is "*the nature of the requirement itself*". Chapter 2 described the state of the art of requirement classifications.

To be more precise, focusing on system requirements, Faisandier (2012) proposes the following system requirement type classification. Figure 60 illustrates the different system requirement types:

1. Functional requirements
2. Effectiveness/performance requirements
3. Interface requirements
4. Utilization or Operational requirements
5. Constraints
6. Validation requirements

In some cases, it exists sub-types, and/or sub-sub-types; for example, for interface requirements, can be distinguished functional interface or physical interface requirements.

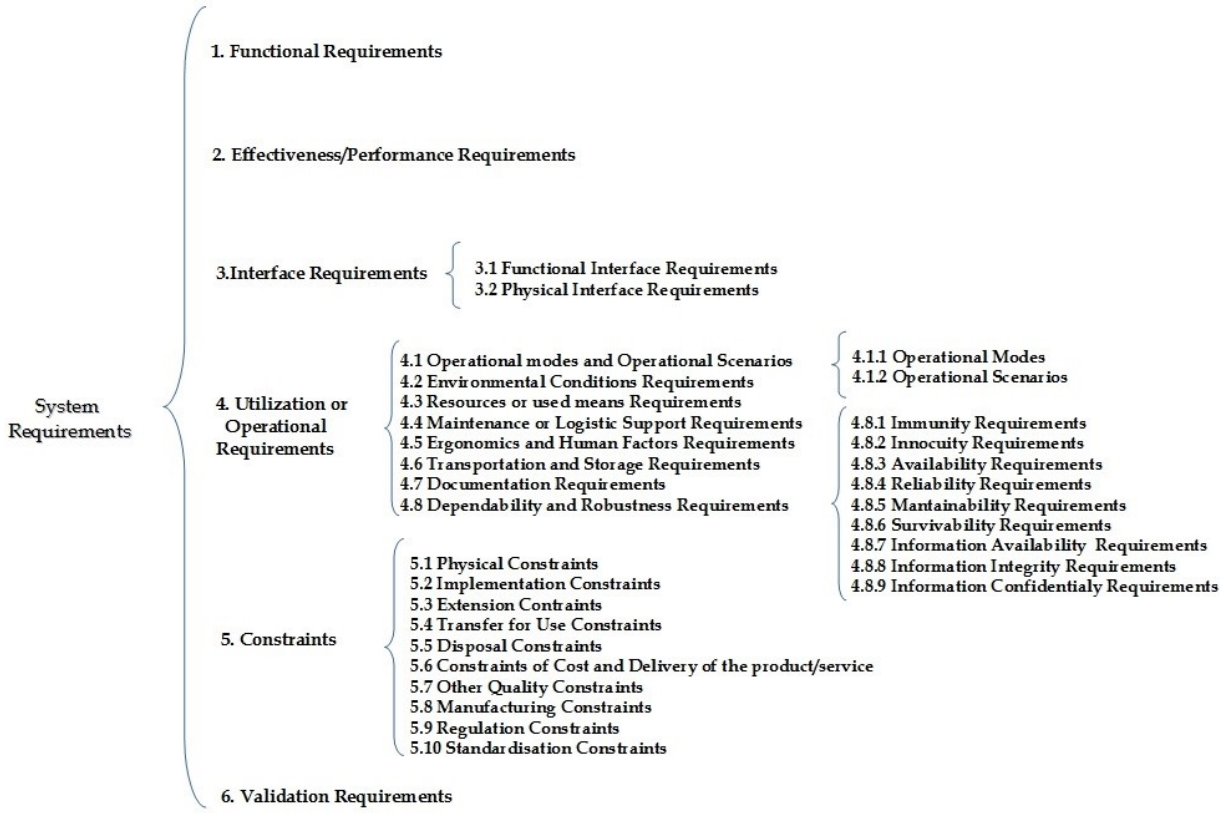


Figure 60 System Requirement Classification according to Faisandier (2012)

It can be observed that there is a synchrony between the *stakeholder requirement type classification* (see Figure 13 Stakeholder Requirement Classification in Chapter 3) and the *system requirement type classification* (Figure 60) proposed both by Faisandier (2012). Table 41 compares both classifications, showing how all categories proposed by the *stakeholder requirement type classification* are analogous to the *system requirement type classification*; in other words, every type of stakeholder requirement can be translated into a certain type of system requirement, keeping this way its traceability.

Table 42 Stakeholder requirement type classification VS system requirement type classification

Stakeholder requirement type classification	System requirement type classification
1. Operational Modes and Operational 1.1 Operational Modes 1.2 Operational Scenarios	4.1 Operational Modes and Operational Scenarios 4.1.1 Operational Modes 4.1.2 Operational Scenarios
2. Expected Services	1. Functional Requirements
3. Expected effectiveness or performances	2. Effectiveness / Performance Requirements
4. Interfaces	3. Interface Requirements
5. Operational Conditions	4. Utilization or Operational Requirements
6. Constraints	5. Constraints
7. Validation Requirements	6. Validation Requirements

Cuiller (2015), ISO 29148 (2011) and SEBoK (2016) also propose a classification of system requirements by type. Table 43 illustrates the names of each category, they are different but cover *almost* the same requirement types proposed by Faisandier (2012); for example, in the left side are listed *design constraints*, *policies and regulations*, and *cost and schedule constraints*, which belong to *constraints* in the right side of the table.

Table 43 System Requirements Classifications by Type and by author

System Requirements Classifications by Type		
SEBoK (2016), ISO 29148 (2011) and Cuiller (2015)	**Faisandier (2012)**	
Name of the categories		
Functional requirements	1	Functional requirements
Performance requirements	2	Effectiveness/performance requirements
Usability requirements	2 6	Effectiveness/performance requirements and Validation requirements
Interface requirements	3	Interface requirements
Operational requirements	4	Utilization or operational requirements
Modes and/or states	4.1	Operational modes and operational scenarios
Adaptability requirements[4]	-	
Physical constraints	5.1	Physical constraints
Design constraints	5	Constraints
Envorinmental conditions	4.2	Environmental conditions requirements
Logistic requirements	4.	Utilization or operational requirements
Policies and regulations	5	Constraints
Cost and Schedule constraints	5	Constraints

[4] Note: *Adaptability Requirements* refers to those system requirements that define potential growth, extension or scalability of the system during its life. Faisandier (2012) only mention "5.3 Extension Constraints" referring to future constraint evolutions.

Activity: Establish the upward traceability with the stakeholder needs and requirements

In order to establish the upward traceability, Ryan & Faulconbridge (2016) suggest the application of Functional Flow Block Diagrams (FFBD) and Requirements Breakdown Structure (RBS), which are briefly described below.

Figure 61 illustrates how the RBS or functional hierarchy begins the definition of requirements from the mission (level 0), goals and objectives to define level 1 system requirements, and continues the downward flow as necessary.

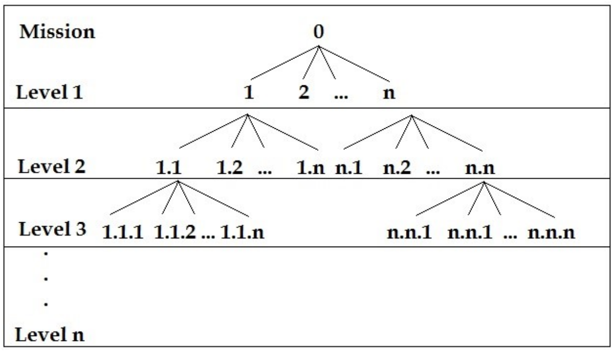


Figure 61 Requirements Breakdown Structure (Ryan & Faulconbridge, 2016)

Figure 62 shows an example of RBS. The system of interest is a domestic security alarm which principal functions shall be deterred unauthorized entry, detect entry, classify entry, report unauthorized entry, and provide a market-leading alarm; each function has been divided into desired sub-functions; for example, the function 'classify entry' is split in: classify entry in un/authorized entry, record classification details, record classifications, indicate false alarm rate, and be able to adjust sensitivity.

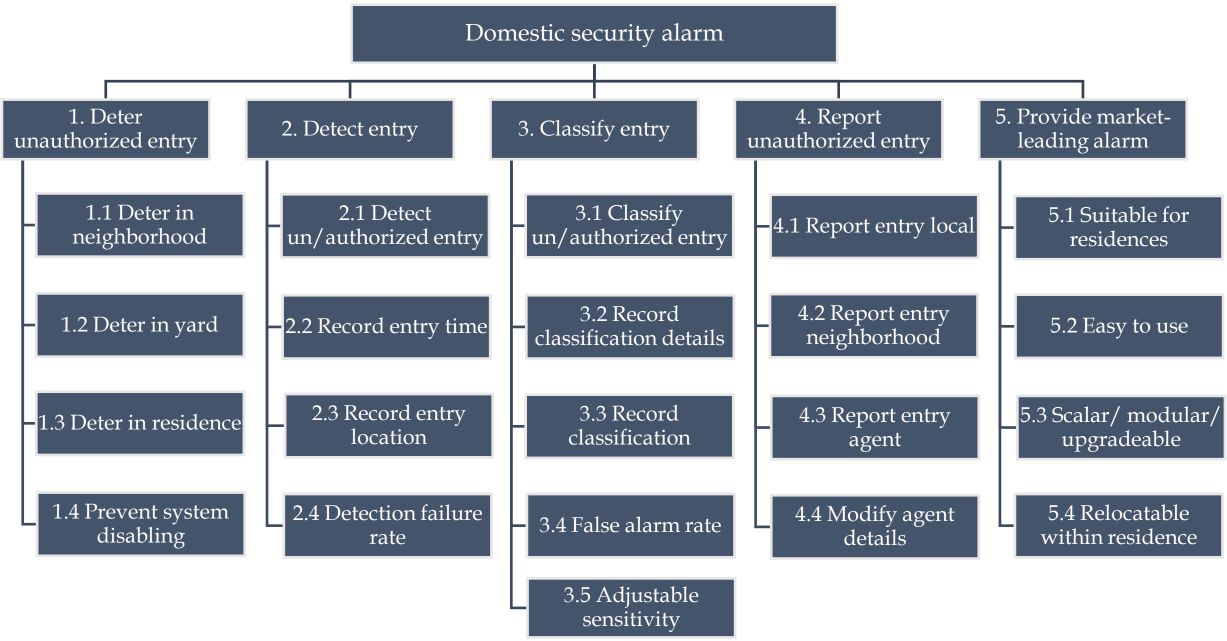


Figure 62 Example of RBS (Ryan & Faulconbridge, 2016)

Functional Flow Block Diagrams (FFBD) are used to support the RBSs diagrams; this kind of diagrams show a hierarchical representation and flow between functions. Figure 63 illustrates the example of the domestic security alarm: the resident sets the alarm and evacuates his home; the alarm is active when the resident departs. If someone entries, the entry may be authorized, unauthorized, or forced entry; in any case, the alarm alerts the entry and, if the resident authenticates, the alarm is inactivated; another way the alarm triggers.

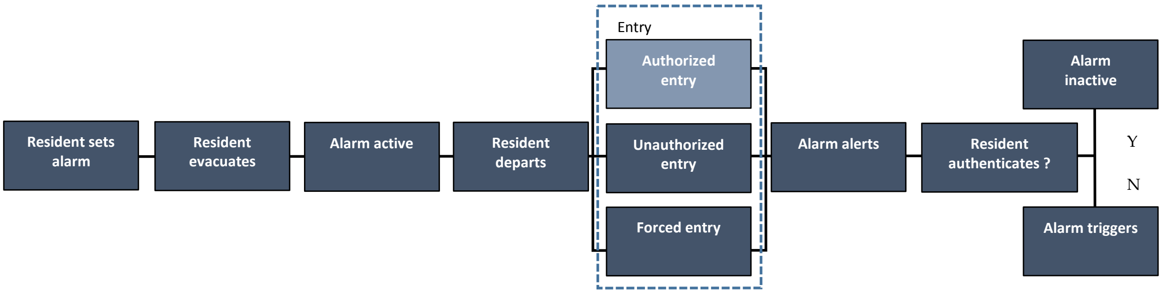


Figure 63 Example of FFBD diagrams (Ryan & Faulconbridge, 2016)

Several authors (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Ryan & Faulconbridge, 2016; SEBoK, 2016; and Cuiller, 2015 for name a few) agree on the need to use automated tools for managing system requirements, since it is practically impossible to trace them without implementing an automated context.

Activity: Establish bi-directional traceability between requirements at adjacent levels of the system hierarchy

SEBoK (2016) recommends the activity: establish *bi-directional traceability between requirements at adjacent levels of the system hierarchy*.

According to ISO 24765 (2017), bidirectional traceability may be defined as the "*association among two or more logical entities that are discernible in either direction (to and from an entity)*" (ISO 24765, 2017, p.45).

Nevertheless, SEBoK (2016) recommends establishing bi-directional traceability between requirements at *adjacent levels* of the system hierarchy. But, what does it means? And, how to find the bi-directional traceability among system requirements at *adjacent levels* of the system hierarchy?

IMPORTANT NOTE: No bibliographic reference, other than SEBoK (2016), was found to explain the term "*bi-directional traceability between requirements at adjacent levels of the system hierarchy*".

Activity: Identify potential risks (or threats and hazards) that could be generated by the system requirements

System requirement risks may be addressed following the same procedures described in Chapter 3, Activity: Identify potential risks (or threats and hazards) that could be generated by the stakeholder requirements.

Activity: Verify the quality and completeness of each system requirement and the consistency of the set of the requirements

Verification can be defined as *"the evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation."* (PMI, 2017).

"Verification involves the comparison between the requirements baseline and the successive refinements descending from it – the product design, detailed design, code, database, and documentation--in order to keep these refinements consistent with the requirements baseline. Thus, verification activities begin in the Product Design phase and conclude with the Acceptance Test. They do not lead to changes in the requirements baseline; only to changes in the refinements descending from it." (Boehm, 1979).

According to Badreau & Boulanger (2014), the objectives of the verification process are:

a) To demonstrate that the initial stakeholder needs have been taken into account,
b) To demonstrate the individual requirements and the set of requirements meet the quality criteria,
c) To demonstrate that no untraceable requirements have been introduced.

Verification of system requirement statement will help to ensure that each of them is expressed correctly and appropriately. Some documents such as ISO 15288 (2015) and Guide for writing requirements (2017) propose that verification be carried out by comparing the quality characteristics of the system requirements against the quality characteristics of a "*good*" system requirement, as well as the quality characteristics of a "*good*" set of requirements. These quality characteristics are presented in Chapter 2.

One quality characteristic of a "*good*" set of requirements is *consistency*. Consistency means that the set of requirements does not have individual requirements that are contradictory (Faisandier, 2012).

As said in Chapter 3, QFD allows to know the co-relationship between the system requirements already defined (Mazur, 2012; Yamashina, Ito, & Kawada, 2002); that is, through this method, it can be known if the different requirements are co-related. For example, if one pair of requirements are co-related in a very strong positive way, it means that if one requirement is changed, the other ones would be affected in the same sense.

Figure 64 illustrates the co-relationship matrix used in the QFD methodology; the system requirements are listed, and after being analyzed, it is symbolized whether every pair of system requirements are co-related or not, in which sense (positive or negative) and in what degree (strong or not).

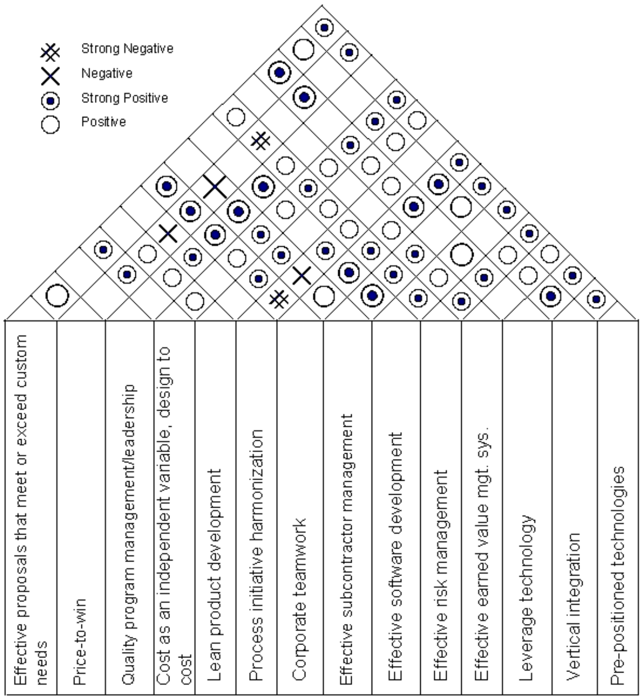


Figure 64 Example of co-relationship matrix between system requirements

For example, from Figure 65, it can be observed that the system requirement *"Process initiative harmonization"* has co-relationship with the system requirements highlighted with arrows. For example, the *'Process initiative harmonization'* has a *positive* co-relation with the system requirement *'Leverage technology '*.

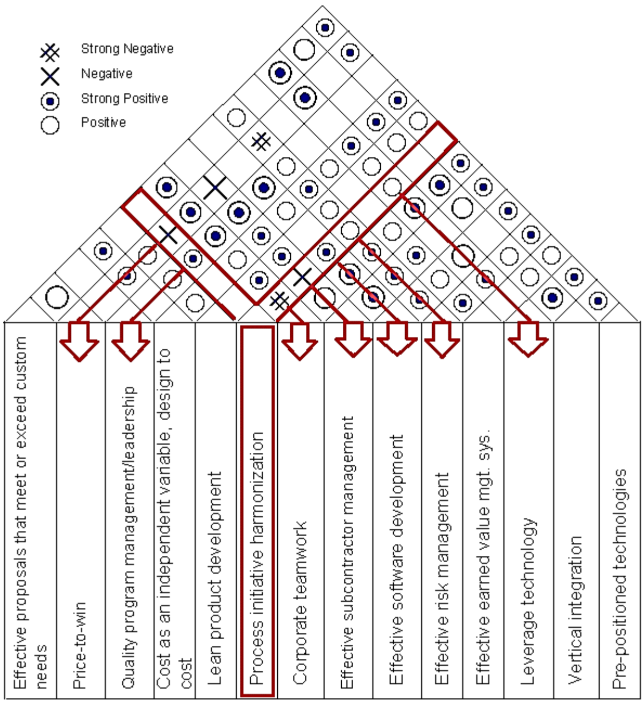


Figure 65 Co-relation of the 'Process initiative harmonization'

Continuing with the example in Figure 65, the *'Process initiative harmonization'* has a *negative* co-relation with the system requirement *'Corporate teamwork'*, and a *strong negative* co-relation with *'Effective subcontractor management'* and *'Price-to-win'*.

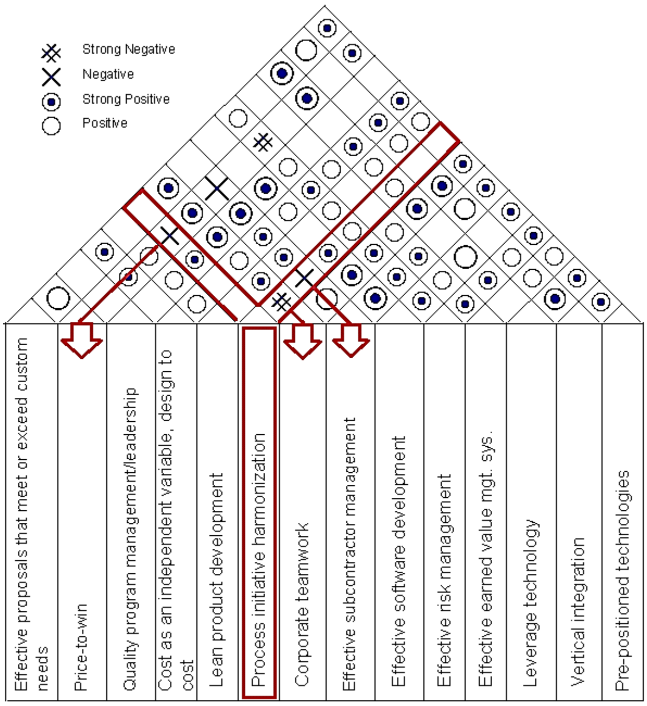


Figure 66 Example of strong negative correlation and negative correlation

At this level of the analysis (system requirements, more technical than stakeholder requirements), it is necessary to reduce or eliminate as much as possible the requirements that are opposite (co-related in a negative or strong negative sense), because this fact could lead to a contradictory design (Kossiakoff et al., 2011). In this perspective, a *negative* co-relation may be seen as a *contradiction.*

Hull, Jackson, & Dick (2011) suggest that classifying requirements in several ways, and the use of filtering and sorting techniques can help to draw together system requirements that share the same topic; when the system requirements are grouped, it will be easier to identify potential conflicts (inconsistencies or contradictions) among them.

Here another problem arises, the analysis of requirements is challenging because of the number of requirements in large system engineering projects; in this perspective, one limitation is the limited human cognitive ability for understanding the whole complexity of the systems and the big volume of data for being analyzed (Mokammel et al., 2018).

Mokammel et al. (2018) propose an automatic approach to requirement extraction, analysis, and graph representation using computational linguistics to identify similarity and contradiction in requirement texts. Table 44 shows the different types of contradiction in system requirement texts; they are antonym, negation, numeric, structure, lexical, and factive, world knowledge.

Table 44 Contradiction in text (Mokammel et al., 2018)

Type	Description
Antonym	**Contradictions exist due to the existence of antonym words. Example:** ***Number of staff should be reduced in the factory. Number of employees should be increased in the factory.***
Negation	**Contradiction exists due to negative words. Example:** ***Temperature in the room should be not superior to 25°C. Temperature in the room should be superior to 25°C.***
Numeric	**Contradictions are present because of different numeric specifications for the system. Example:** ***The weight of the handset should be lower than 113 g. The weight of the handset should be lower than 200 g.***

Structure	The physical configuration of the system contradicts with the requirements. Example: *An Internet submarine cable link should be constructed between the Czech Republic and Finland.* (Impossible because of lack of water connection between the two countries).
Lexical	Lexical or semantic discrepancy leading to a contradiction. Example: *All the system's components should be manufactured locally. Batteries should be imported from our German supplier.*
Factive, world knowledge	Contradiction generated by the lack of consideration of establishing knowledge. Example: *Sand from the Sahara desert will be used to build the bridge.* (This is impossible because Sahara sand cannot be used in construction work due to the round shape of the particles).

Figure 67 illustrates how this tool helps to *cluster* system requirements by grouping those ones that show similarity in text: the cluster or group is represented by a specific color and a specific type of line; here, cluster 1 is represented in blue and dotted line, cluster 2 in red and continuous line, cluster 3 in green and discontinuous line. The symbol X represents the center of the cluster. For example, it can be observed that requirements 13, 14, 15, 16, 17 and 18 are grouped in cluster 1 (blue, dotted line).

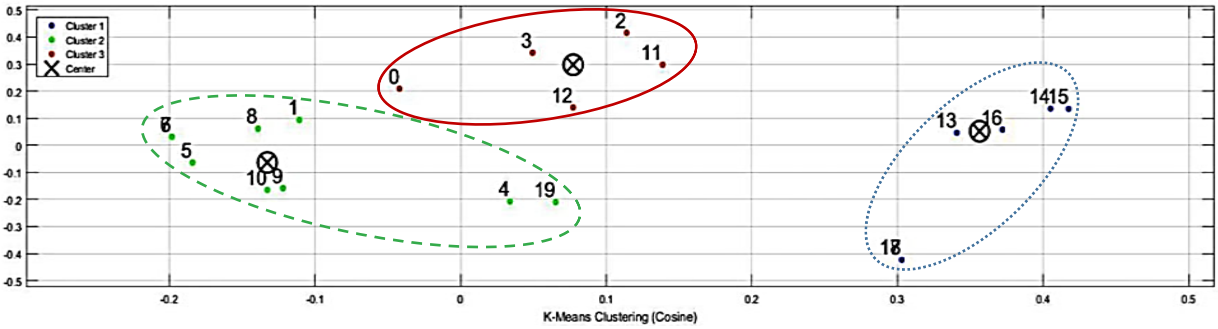


Figure 67 Clustering for the system requirements (adapted from Mokammel et al., 2018)

Once the clusters are identified, the software creates links between *similar requirements* and *contradictory requirements*. Figure 68 Graphic representations of requirement relation (Mokammel et al., 2018) illustrates the links (similarity and contradictory) of the example presented above: potential similar requirements are linked with a continuous line, while potential contradictory requirements are linked with a non-continuous line. For example, requirement 7 (Req 07) has a potential contradiction with requirement 6 (Req 06) and potential similarity with requirement 1 (Req 01); note that requirements 1, 6, and 7 belongs to the same cluster "green".

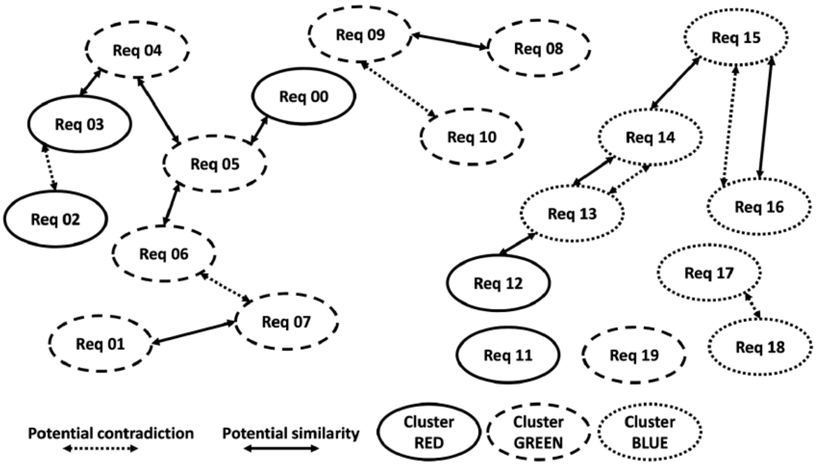


Figure 68 Graphic representations of requirement relation (Mokammel et al., 2018)

Once contradictory system requirements are identified, how can they be treated?

Due to the difficulty of satisfying both negative co-related system requirements, Hauser & Clausing (1987) explain that "*it is not an easy decision*" when two system requirements are negative co-related, the stakeholders should take decisions based on objective competitors data, costs, technical difficulties, and customer perceptions.

When taking decisions, Kossiakoff et al. (2011) point out the relevance of *cost* criteria; maybe, the amount of investment necessary to develop and produce the system would result in a per-unit cost much more superior than competing systems, and finally, it will not be a success. In this context, it is worthy to identify if this is a *key requirement*. Every key requirement should have a negative answer to the questions:

- *"If the solution did not provide me with this capability, would I still buy it?*
- *If the system did not do it, would I still want it?"* (Hull, Jackson, & Dick, 2011)

If the negative co-related system requirements are key requirements, it is worthy to try to solve the contradiction between system requirements.

Authors like Malinin (2016) or Kraev (2006) identify two general types of contradictions:

> **A technical contradiction** is "*a conflict between characteristics within a system*" where the improvement of one parameter leads to the worsening of another (Kraev, 2006).
>
> **A physical contradiction** is "*a conflict between two mutually exclusive physical requirements to the same parameter of an element of the system*" (Kraev, 2006).

This way, technical contradictions are typically related to properties of the whole technical system, and physical contradictions are related to physical properties of one quality characteristic of an element of the system. Table 45 shows examples of technical and physical contradictions; for example, if power is improved, weight is worsened; if the system is liquid, the system can-not solid.

Table 45 Examples of technical or physical contradictions (Kraev, 2006)

Technical contradictions		Physical contradictions	
Improving parameter	**Worsening parameter**	**Characteristic "A"**	**Characteristic "Non-A"**
Power	Weight	Electro-conductivity	Dielectric
Complexity	Functionality	Liquid	Solid
Adaptability	Reliability	Hard	Soft
Productivity	Precision	Fast	Slow
Convenience of use	Manufacturability	Strong	Weak

Once those contradictory requirements have been identified, Kraev (2006) suggest not to avoid contradictions, instead of this, it recommends to intensify them. The author proposes two methodologies to resolve contradictions:

1. To apply the 40 Inventive Principles for overcoming the contradiction (Altshuller, 2000). Note: this suggestion is also supported by Yamashina, Ito, & Kawada (2002).

Or

2. To transform the technical contradiction into physical contradiction and solve the conflict on the physical level. The steps are the following and the methodology is illustrated with examples taken from Kraev *(2006):*
   A. ***Identify the opposing technical contradictions***.

B. ***Describe the technical contradiction.*** Altshuller (2000) identified 40 characteristics useful to develop and describe a technical contradiction. Table 46 shows some technical contradiction, for example: if a nail is improved in fixation, its manufacturability is worsened.

Table 46 Improving or Worsening Parameter in Technical Contradictions (Examples) (Kraev, 2006)

Problem name	Technical contradiction	
	Improving parameter	Worsening parameter
"Nail"	Fixation	Manufacturability
"Box"	Sliding	Design complexity
"Refrigerator"	Information	Loss of time

C. ***Transform the technical contradiction into physical contradiction.*** Almost all technical contradictions may be transformed into physical contradictions; this can be done through the definition of a specific physical problem that may be solved with aid of the four "physical principles" (separation in time, separation in space, system transformations, and phase transformations, or physical-chemical transformation of substances). Table 47 illustrates three examples of physical contradictions; the nail's rod profile can be circular or non-circular, but not both; the box' guide surface can have low friction or high friction; and the refrigerator's indicator "should be" or "should not be".

Table 47 Examples of physical contradictions (Kraev, 2006)

Problem name	Parameter	Physical contradiction	
"Nail"	Rod profile	Circular	Non-circular
"Box"	Guide surface	Low friction	High friction
"Refrigerator"	Indicator	Should be	Should not be

D. ***Separate the opposing physical contradictions and remove the combined area of their interaction.*** If the opposing requirements are separated in time, system, and space, they should not have a superposition (conflict). Figure 69 shows how the box (b) and the refrigerator (c) have separated areas (in time, space, or system), illustrating that the contradiction does not exist anymore; case contrary to nail (a), that seems to have a certain combined area where the inconsistency remains.

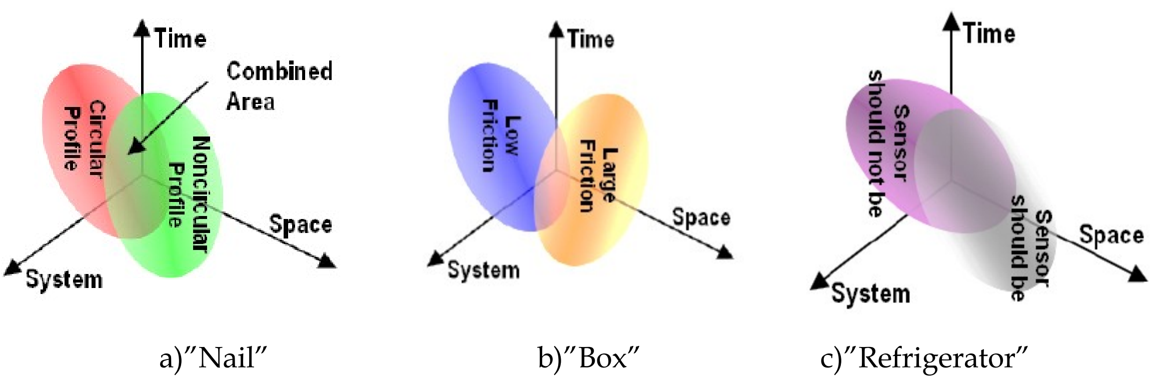


Figure 69 Separation of the Opposing Physical Contradictions (Kraev, 2006)

E. ***Obtain solutions.***

Another proposition to resolve contradictions is done by Malinin (2016) who proposes a methodology that considers: a) the different locations of the system, b) the different life stages of the system, and c) the different conditions of the system at different times. His methodology has three steps and it is illustrated here with an example (system: medical catheter) taken from Malinin (2016):

I. **Restatement of the initial problem to better understand the requirements.** Example: "Does the whole catheter need to be radio-opaque?" and "Does the whole catheter need to be alcohol resistant?" In this example, the requirement considers different locations of the system. See Table 48, in the proximal part of the system it is very important that the catheter is alcohol resistance, but it does not matter if it is radio-opaque (visibility it is not important); on the other hand, in the distal part it is very important that the catheter allows visibility (radio-opaque) and it does not matter the alcohol resistance. In addition, at the inner part of the system it is not important to have alcohol resistance or visibility; contrary, at the outer diameter it is very important the alcohol resistance, and medium important (1) the visibility.

Table 48 Requirements for different parts of the catheter (Malinin, 2016)

	Proximal	Distal	Inner diameter (ID)	Outer diameter (OD)
Alcohol resistance	2	0	0	2
Visibility	0	2	0	1

Where: 0 = Not important, 2 = Important

II. **Formulating a contradiction as a technical contradiction or physical contradiction**. Example:
   a. Technical contradiction: "If radio opaqueness of the catheter improves, its alcohol-resistance deteriorates".
   b. Physical contradiction: "the level of radio-opaque filler in the catheter needs to be high (for radio opaqueness) and at the same time the level of radio-opaque filler needs to be low (for alcohol resistance)".

III. **The solution of the reframed problem using technical or physical contradiction.**
   a. Using the technical contradiction: it is found through the expression of the contradiction, the recommendations found in the generalized contradiction matrix, and finally applying the 40 Inventive Principles (Altshuller, 2000). Example: "If radio opaqueness of the catheter improves, its alcohol-resistance deteriorates. By mapping the contradiction expressed in specific terms to the generalized contradiction matrix, the generalized contradiction can be stated as: if "Difficulty of detecting and measuring improves", then "Strength" deteriorates. The respective inventive principles, from the matrix, are: Cheap Short-living; Local Quality; Dynamics; Mechanics Substitution".
   b. Using the Physical contradiction: to solve it, it is necessary to formulate two mutually exclusive contradictory requirements. Example: "Which parts of the catheter need to meet which requirements?" To meet the requirements, the catheter may be manufactured varying its properties along its length.

But, what to do if after applying the suggestion of Kraev (2006) (time, system or space solution) or Malinin (2016) (the different locations of the system, the different life stages of the system, or the different conditions of the system at different times solution) it still remains a conflict among requirements? For example, the one shown in Figure 69 a) where there is a *combined area* or *superposition* that increases the difficulty of finding a solution.

In this case, as stated in Chapter 3, it is recommended to negotiate and find consensus among involved stakeholders, in order to avoid the risks of contradictory design in the development of the system.

Activity: Validate the content and relevance of each system requirement against the set of stakeholder requirements

Validation can be defined as *"the assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification."* (PMI, 2017).

According to Boehm (1979) *"Validation identifies problems which must be resolved by a change of the requirements specification… For example, a simulation of the product design may establish not only that the design cannot meet the baseline performance requirements (verification), but also that the performance requirements are too stringent for any cost-effective product designs, and therefore need to be changed (validation)".*

Kossiakoff et al. (2011) express that system requirement validation may be accomplished *formally* or *informally*. *Formal validation* implies the participation of an external entity that, with the aid of several validation methods, will or will not validate the set of system requirements against operational situations; these validations will determine or not if in the requirements embodied within a system concept could achieve the user needs. *Informal validation* refers to review the set of system requirements with the customers and/or users to determine the scope, completeness, and comprehensiveness of the requirements.

Badreau & Boulanger (2014) explain that validation process may be performed by formal activities such as concept document review, prototypes and/or models, and the preparation of the *system acceptance test booklet* for the customer.

Faisandier (2012) and SEBoK (2016) suggest the use of the validation traceability matrix that relates stakeholder requirements to system requirements; this matrix will help validate - in content and relevance - that each stakeholder requirement has been translated into at least one system requirement. They suggest including the rationale for each system requirement.

SEBoK (2016) suggests early prototyping for the identification of functional, operational and system constraints. Prototypes improve user understanding of the system requirements by allowing interaction, discovery, and feedback; this increases the likelihood that the system will be accepted by the user when its needs are met.

SEBoK (2016) lists as main activities of the validation process:

- Establishment of a validation strategy
- Carrying out the validation
- Analysis of the results obtained and their comparison with the expected results

- Process control (perform corrective actions based on detected non-conformities reports)

Several authors (Blanchard & Fabrycky 2006; Faisandier 2012; SEBoK 2016, Ryan & Faulconbridge 2016) agree that each system and each element of the system must be verified and validated, and possibly corrected before it can be integrated into a higher level *parent* system.

Graphically, SEBoK (2016) outlines the verification and validation processes by system level as follows (see Figure 70). In the left side of the diagram, they can be seen the stakeholder and system requirements, by levels; the fulfillment of system requirements should be verified against the stakeholder requirements, to be sure that *the system is constructed right*. In the right side, they can be seen the systems or products, by levels, that should be validated against the requirements from the left side, to be sure that *the right system is constructed*.

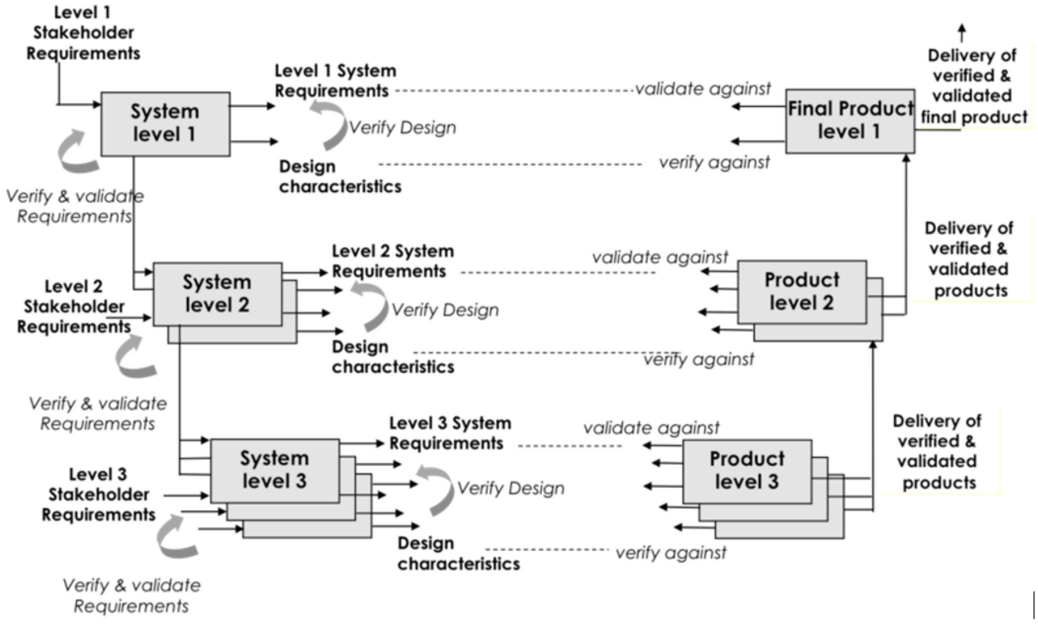


Figure 70 Verification and Validation Level per Level (SEBoK, 2016)

According to SEBoK (2016), throughout the whole life cycle, the *users* of the system must be considered and involved in validation, to ensure that all aspects of user needs are prioritized and taken into account.

Blanchard & Fabrycky (2006), Schillo, Isabelle, & Shakiba (2017) and Hauser & Clausing (1988) state that the Quality Function Deployment (QFD) technique is a powerful tool for selecting requirements and also for comparing design features (system requirements) against user needs. Kossiakoff et al. (2011) propose a scale related with numerical values to establish the relationship between stakeholder requirements and system requirements; the scale is: strong positive (9), medium positive (3), weak positive (1), or negative (-3). See Figure 71. In the frame of QFD, these values are used to carry out market competition studies and determine if the new system is better, equal or worse than those systems already in the market.

		Voice of the Company													
		Enterprise Product Development Capabilities													
		Business Capture		Quality Processes				Project Management					Technology Development		
⦿ Strong Symbol 9 △ Weak Symbol 1 ○ Medium Symbol 3	Customer Importance	Effective proposals that meet or exceed custom needs	Price-to-win	Quality program management/leadership	Cost as an independent variable, design to cost	Lean product development	Process initiative harmonization	Corporate teamwork	Effective subcontractor management	Effective software development	Effective risk management	Effective earned value mgt. sys.	Leverage technology	Vertical integration	Pre-positioned technologies
Direction of Improvement		↑	↑	○	↑	↑	↑	↑	○	↑	↑	○	↑	○	○
Customer Demanded Quality: Produce innovative solutions that work	0.167	⦿	⦿		⦿			⦿		○			⦿	⦿	⦿
Quality of product	0.167			⦿	⦿					⦿	○		⦿		⦿
Company that is open, honest, understanding of customer needs	0.167	⦿						○	⦿		⦿				
Effective customer contact	0.167	⦿			⦿			⦿						○	
Want products on-time	0.167						○	○	⦿	⦿		○			⦿
Target program cost performance	0.167		△		⦿	⦿			⦿		○	○			⦿

Figure 71 Example of relationship matrix between stakeholder requirements and system requirements

Boehm (1979), suggests that verification and validation (V&V) activities should be performed by three different roles: a) V&V agent, b) specification agent, and c) project manager and customer. The author state that these activities should be iterative between the system requirements and design specifications, in which:

- *"The V&V agent analyzes the specifications and issues problem reports to the specification agent;*
- *The specification agent isolates the source of the problem, and develops a solution resulting in an iteration of the specification;*
- *The Project Manager and Customer approve any proposed iterations which would perceptibly change the requirements baseline;*
- *The V&V agent analyzes the iterated specification, and issues further problem reports if necessary;*
- *The process continues until the V&V agent completes his planned V&V activities, and all problem reports have been either fixed or assigned to a specific agent for resolution within a given time."* (Boehm, 1979).

Activity: Synthesize, record, and manage the system requirements and potential associated risks

Faisandier (2012) proposes that once resolved the conflicts between opposite system requirements, similar requirements should be synthesized. SEBoK (2016) and Cuiller (2015) express that synthesize the system requirements will carry some benefits as reducing the overall number of system requirements, exposing incorrect assumptions of designers in early stages, eliminating design implementation as a requirement, and improving communication between stakeholders and the design team by capturing rationale requirements. The reduction in the total number of requirements may be done through the identification of *duplicates*, and this reduction will reduce project cost and risk (SEBoK, 2016).

As a result of the process of defining system requirements, we will have artifacts such as:

- System Requirements Document
- System Requirements Justification Document (for traceability purpose)
- System Requirements Database, including traceability, analysis, rationale, decisions, and attributes, where appropriate.
- System External Interface Requirements Document (this document describes the interfaces of the system with external elements of its context of use) the interface requirements can be integrated, or not, to the system requirements document.

The system requirements prepare the necessary inputs for the architecture and design activities; they will also serve as a reference for the validation and justification of the implemented system.

Ryan & Faulconbridge (2016) mention that requirements management is the process that manages changes in system requirements throughout the system's lifecycle. These changes in system requirements are due to various causes such as:

- These requirements are derived from stakeholder requirements and are managed as they are developed.
- Changes in stakeholder requirements
- Changes in the business
- The environment can change throughout the life of the system.
- Laws and regulations may change as the project develops

Ryan & Faulconbridge (2016) state that more than 50% of the system requirements are modified before the system is put into service. These authors assure that requirements cannot be properly managed without traceability.

There are many tools that assist the Requirements Engineering for the administration of such requirements, Ryan & Faulconbridge (2016) mention among them:

- the context diagram
- functional flow block diagrams
- requirements breakdown structure
- N2 diagrams
- structured analysis
- data flow diagrams
- control flow diagrams
- behavior diagrams
- action diagrams
- quality function deployment
- state/mode diagrams
- process flow diagrams
- function hierarchy diagrams
- state transition diagrams
- entity relationship diagrams
- structured analysis and design
- object-oriented analysis
- unified modeling language
- structured systems analysis
- design methodology

Nevertheless, processes, methods, modeling techniques and systems engineering ontology are only partially supported by computer tools, like those that exist allow the exchange between the internal actors of the project, and exchanges can seldom take place between the client and the supplier (Faisandier, 2014).

One of the greatest difficulties encountered in an interdisciplinary and inter-organizational context lies in the exchange of information between the different computer tools that support systems engineering; a second difficulty relates to the perennial nature of information and models that must "live" the time the system "lives" (Faisandier, 2014).

According to Cuiller (2015), the most widely used tools for administration and traceability of system requirements are Word and Excel; however, there are some important disadvantages such as complexity when the system has several thousand requirements, nor do they provide support for change management. According to his experience in consulting, the most widespread tool for requirements management is IBM's Rational DOORS.

As regard to the tools, one can currently find computer tools on the market for requirements management, among them we can mention: Rational DOORS Next Generation by IBM, OSLC y Context® SMD by Mentor Graphics, ATEGO Process Director, or CAPELLA for example.

Ryan & Faulconbridge (2016) suggest that the automated requirements management tool should be capable of:

- Provide support during elicitation of stakeholder requirements
- Enable readers to explore, retrieve specific requirements to be retrieved, and generate requirement reports from specific search criteria
- Support forward and backward traceability
- Support in the generation of requirements that meet the characteristics of a good requirement
- Enable the export and import of requirements in various formats such as word processors or spreadsheets
- Support in change control and the corresponding evaluation
- Support functional localization and translation of functional-to-physical.
- Does not impose any requirements engineering process

Cuiller (2015) precises his own expectation regarding a tool for requirements management:

- Store and retrieve requirements documents from all levels of development
- Edit and store requirements with their appropriate attributes
- Support validation and verification of requirements
- Establish traceability links between requirements, as well as between requirements and validation and verification elements.
- Support change management
- Keep a history of requirements
- Publish requirements documents

According to Badreau & Boulanger (2014), the document where the requirements are specified should allow the verification of these complementary criteria: structure, modularity, extensibility, sufficient, and traceability.

4.3 Discussion

After reviewing literature, it can be said that authors like Hull, Jackson, & Dick (2011), Faisandier (2012), SEBoK (2016), Ryan et al. (2015), ISO 15288 (2015), and Ryan & Faulconbridge (2016) agree in the activities to perform in order to translate stakeholder requirements into system requirements.

As said in Chapter 3, there is a large amount of information regarding how to conduct these activities and related tasks. In addition, it remains the opportunity to design a lean process to increase value and reduce waste.

4.4 Conclusion

Chapter 4 presented the state of the art of how to translate stakeholder requirements into system requirements.

The different activities to translate stakeholder requirements into system requirements were identified. They are: analyze stakeholder requirements to verify completeness, transform stakeholder requirements into system requirements defining their rationales, incorporating the derived and decomposed requirements (coming from architecture and design) into the system requirements baseline, classify the system requirements according to their type, establish the upward traceability with the stakeholders needs and requirements, establish bi-directional traceability between requirements at adjacent levels of the system hierarchy, identify potential risks (or threats and hazards) that could be generated by the system requirements, synthesize, record, and manage the system requirements and potential associated risks, verify quality and completeness of each system requirement and the consistency of the set of the requirements, and validate the content and relevance of each system requirement against the set of stakeholder requirement.

Chapter 5 will present the lean thinking fundamental concepts, the state of the art of the lean enablers for system life-cycle phases, and how *lean* is applied in quality management systems.

CHAPTER 5 Lean Thinking

5.1 Introduction

As it has been said before, the general objective of this research work is focused on the development of a methodology and its tools, based on Systems Engineering and in compliance with the standard ISO 15288 (2015), to ensure, as far as possible, that the stakeholder needs are understood, and later, are translated into system requirements, reducing risks of project abortion or delays, and adding *value* to the analysis and design process.

This objective can be achieved through getting inspiration from different domains, techniques, and methodologies. In this research work, Quality Assurance, Systems Engineering, and Lean Thinking are the domains that interact in order to meet the general objective:

- Systems Engineering deals with the system life cycle processes (stakeholder needs identification, stakeholder and system requirements definition);
- Quality Assurance monitors these processes to error prevention and requirement fulfillment, as well as generates documentation for quality managing purposes; and
- Lean Thinking philosophy will reduce risk of project abortion, adding value to the processes while reducing waste.

First, Chapter 1 developed the problem context and research focus taking into account the standards ISO 9001 (2015) *Quality Management Systems – Requirements*, and ISO 15288 (2015) *Systems and Software Engineering - System Life Cycle Processes*. Later, Chapter 3, and Chapter 4 described how Systems Engineering addresses the problem. Finally, this chapter now details how Lean Thinking philosophy has influenced Systems Engineering and Quality Assurance to add value to both domains.

In this chapter the reader will find the fundamental concepts of Lean Thinking, how this philosophy has been applied to enable *lean* into the system life cycle phases (Systems Engineering) and also how *lean* has been implemented in quality management systems (Quality Assurance). This chapter will teach us the essentials for developing a *lean* methodology and its tools, with value-added and waste reduced, as well as the good management practices to reduce risk of project abortion or delays. We will apply the concepts learned here to fulfill the general objective of this research work: develop the methodology and its tools (Chapter 6).

5.2 Lean Thinking fundamental concepts

Lean Thinking concept emerged in Japan during the last century in a Japanese automobile manufacturing company: Toyota Motor Company; Toyota broadly applied the Taylor Science

Management System steps to manufacturing, cost of sales, administrative and capital costs, creating the concept of lean manufacturing (Niebel & Freiwalds, 2009).

In accordance with Oehmen (2012), the activities of any process are classified into three categories depending on the value they add to the process of which they form part:

a. Value-added activities: transform information or material, or reduce uncertainty; is an activity that the customer would approve; the activity is done right the first time;
b. Required or necessary non-value-added activities: activities that cannot be eliminated because they are required by law, contract, or other similar reason;
c. Non-value-added activities: these activities consume resources and create no value.

In manufacturing, lean process concepts are focused on process flow and on reducing the number of non-value-adding, flow-impeding activities. The opposite of a lean process is a process in which work does not flow and there are side passes, bottlenecks, waiting times, high inventories, numerous activities that are done by routine and tradition, but that do not add value to the product (Gutierrez Pulido, 2010).

Bauch (2004) comments that there is the possibility of transferring the principles applied in manufacturing to the product development area. Similarly, Baines et al. (2006) assure that the lean concept is applicable to the Product Design process.

Therefore, lean manufacturing concepts can be adopted during the Product Analysis and Design phases in order to reduce waste and preserve value-adding activities. The design of the new product, or system, should in turn support manufacturing processes to consider only value-adding activities (lean design for lean manufacturing) (Bauch, 2004; Baines et al., 2006).

Starting from the comments of Baines et al. (2006), it is a reality that there is currently no slender product design process, which only contains activities that add value to it; the authors point out that the redesign of the process flow continues being an unresolved issue.

For Bauch (2004), the four value streams that must be considered when analyzing and designing a system are: 1) compliance with customer requirements, 2) prototype design and validation, 3) product manufacturing, and 4) necessary procurement. Focusing on these four areas, the analysts and designers may reflect on the *value* of the system being designed, as well as the *waste* to be avoided. *Value-stream-map* is a tool that aids the visualization of the value stream in processes.

Figure 72 represents a value-stream-map that illustrates the first two value streams recommended by Bauch (2004); the example is an information system development process. The figure shows: 1) compliance with customer requirements, and 2) prototype design and validation. In the top of the image are arranged chronologically the system development steps (data modeling, coding, integration testing, and user acceptance testing); in the left side, there are the activities classified depending on the value they add to the process. The diagram represents the time consumed for each activity through the system development process.

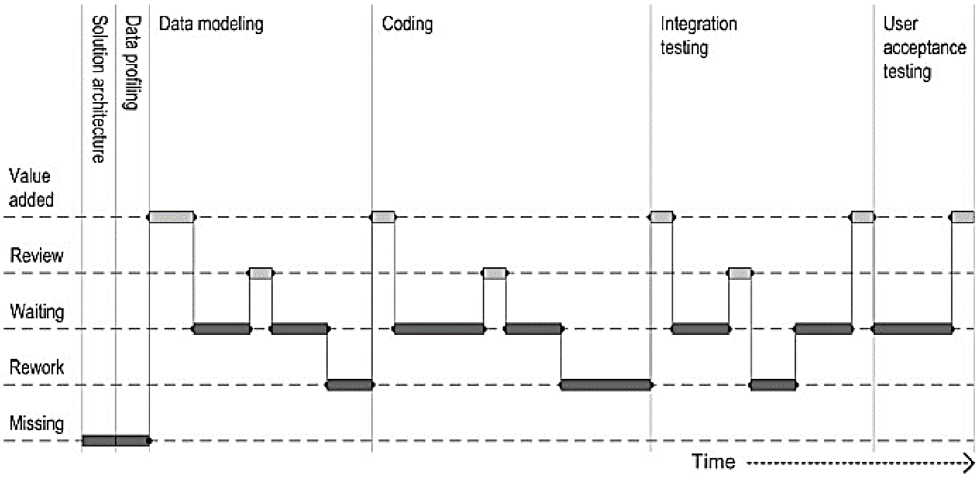


Figure 72 Value-stream analysis for an information system development (Hughes, 2016)

Lean Thinking has three fundamental concepts: *waste*, *value*, and *creating value without waste*. Next subsections precise these concepts.

Waste

Considering a process as the object of study, anything or any activity that generates costs but does not add value to the product is considered a *waste* or *muda* (Gutierrez Pulido, 2010). Walton (1999) describes two classes of muda:

a. Type I muda: found in activities that add no value to the client, but are necessary to deliver the system, for example: inspection/review/verification (required or necessary non-value-added activities according to Oehmen, 2012); and
d. Type II muda: found in activities that do not create any value and can be eliminated immediately, for example: transportation (non-value-added activities according to Oehmen, 2012).

Ohno (1988, cited in Walton, 1999; Gutierrez Pulido, 2010), identified seven forms of waste found in physical manufacturing:

1. Over-production (ahead of demand)
2. Inventory (more than absolutely necessary)
3. Transportation (of materials)
4. Unnecessary movement (of operators during their activities)
5. Waiting (for the next process)
6. Defective products (no-quality/non-conforming products)
7. Over-processing (of parts due to poor tool and product design)

Walton (1999) and Bauch (2004) applied the seven forms of waste in manufacturing to *product development*. In fact, Bauch (2004) adds three more new waste driver categories in product development. Table 49 shows the contributions of both authors regarding the wastes in product development; for example, Walton (1999) considers for the type of waste *over-production*; the identified wastes in product development are: *too much detail, unnecessary information, redundant development (reuse not practiced*).

Table 49 Seven to ten types of waste applied to product development

Type of waste	Wastes in product development	
	Walton (1999)	Bauch (2004)
Over-production	Too much detail Unnecessary information Redundant development (reuse not practiced)	Poor synchronization as regards contents Poor synchronization as regards time and capacity Over-dissemination of information Redundant tasks
Inventory	Too much information Incomplete content Poor configuration Management	Unnecessary testing equipment and prototypes Excessive data storage Critical path related queues High system variability Exceeding capacity utilization Large batch sizes
Transportation	Information/Software incompatibility Communications failure Not standards-based Multiple sources Incompatible destinations requiring multiple transports	Excessive data traffic Hand-offs Stop and go task / Task switching Ineffective communication
Unnecessary movement	Information user not connected to sources requiring manual intervention Information pushed to wrong people	Lack of direct access to information Information hunting Remote locations
Waiting	Information created too early Late delivery of information Unavailable information Quality suspect	Unavailable information Unavailable manpower or computing resources
Defective products	Quality lacking or suspect Conversion error Wrong level of information Incomplete information Ambiguous information Inaccurate information Tolerance exceeded Poor configuration management	Deficiencies in information quality (IQ) attributes Erroneous data and information Poor testing and verification
Over-processing	Unnecessary serial processing Lack of needed information Poor/bad decisions affecting future Excess/custom processing Not processed per process Too many iterations/cycles Unnecessary data conversions Excessive verification No transformation instructions Decision criteria unclear Working with the wrong level of detail Propagation of bad decisions Processing of defective information Multiple tasking when not required	Unnecessary features and processes Unnecessary detail and accuracy Excessive approvals Excessive transactions Inappropriate use of competency Use of inappropriate tools/methods
Re-invention		Poor design re-use Poor knowledge re-use
Lack of system discipline		Unclear goals and objectives Unclear roles, responsibilities, and rights Unclear rules Poor schedule discipline Insufficient readiness to cooperate Incompetency/poor training
Limited information technology resources		Poor compatibility Poor capability Low capacity (availability)

After analyzing the content of Table 49, the type of waste "*Re-invention*" (Bauch, 2004) is contained in the type of waste "*Over-production*" (Walton, 1999) waste "*Redundant development (reuse not practiced)*". In addition, types of waste "*Lack of system discipline*" and "*Limited* *information technology resources*" (Bauch, 2004) may be considered as *weaknesses* of the company that leads to waste. The weakness "*Lack of system discipline*" can be solved through methodologies such as the Balanced Score Card, a methodology that helps to overcome the dispersion or duplication of efforts, allows the creation of synergies, supports the daily operation, and is connected with the mission, vision and strategic objectives. However, this strategy requires to be understood and assumed by all (Gutierrez Pulido, 2010). The weakness "*Limited information technology resources*" can be solved if the technology is available, and if the company is able to invest in it.

In any process, waste can be identified with the aid of tools like value-stream-maps (Tyagi et al., 2015; Darwish et al., 2016; Jeong & Yoon, 2016) or flow diagrams (Niebel & Freiwalds, 2009). Once the waste is identified, it can be reduced or eliminated, resulting in an improvement for the process. Clearly, product development processes can be improved through the reduction or elimination of these forms of waste.

This way, continuing with the example of the information system development, Figure 73 illustrates the *lean* product development process. There were removed the non-value-added activities such as *waiting* and *rework*, and there were kept the value-added activities and required-non-value-added activities (*review*). The result: a *lean* product development process for information system development.

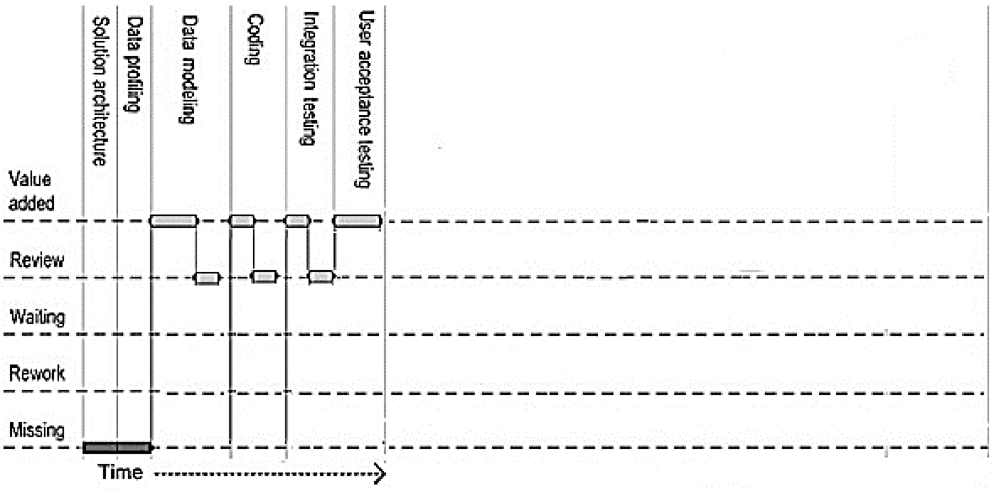


Figure 73 Value-stream analysis after lean (adapted from Hughes, 2016)

According to Hohmann (2019) and Jeong & Yoon (2016), analyzing product development processes through value-stream-maps (lean principles) may represent improvements by as much as 40% lead time reduction, depending on the nature of the project. Figure 74 illustrates the lead time reduction of the information system development process example; the arrow above shows the total process time *before lean*, the arrow below represents the total process time *after lean*.

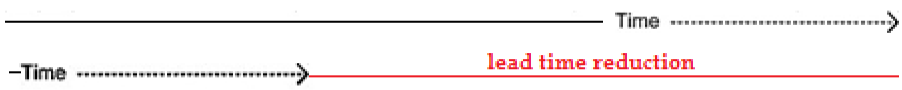


Figure 74 Lead time reduction after applying lean

Value

Considering a process as the object of study, as it has been said, a *value-added activity* is an activity that adds value to the process through transforming information or material, or reducing uncertainty (Oehmen, 2012).

Nevertheless, the term *value* has a much wider application. *Value*, considered from the point of view of the product, is what the customer considers important, and is willing to pay for it (Oehmen, 2012). Talking about the project, the *value* is determined in terms of the expectations and benefits desired by each stakeholder (Jepsen & Eskerod, 2009).

De Weck & Willcox, (2010) describe how *value* is perceived by different stakeholders. For example, these authors make a difference between *customer value* from *shareholder*[5] *value*. Figure 75 illustrates how customer value is derived from quality product, timeliness, and price, and the shareholder value is derived from the economic value added. The economic value added results from subtracting costs from revenues. Nevertheless, even if their perception of value is different, the quality product, timeliness, and price (*customer value*) remain to be the most important factors to satisfy. This is because the revenue (*shareholder value*) is directly related to customer satisfaction, if the customer is satisfied its demand will increase, and the revenue will increase.

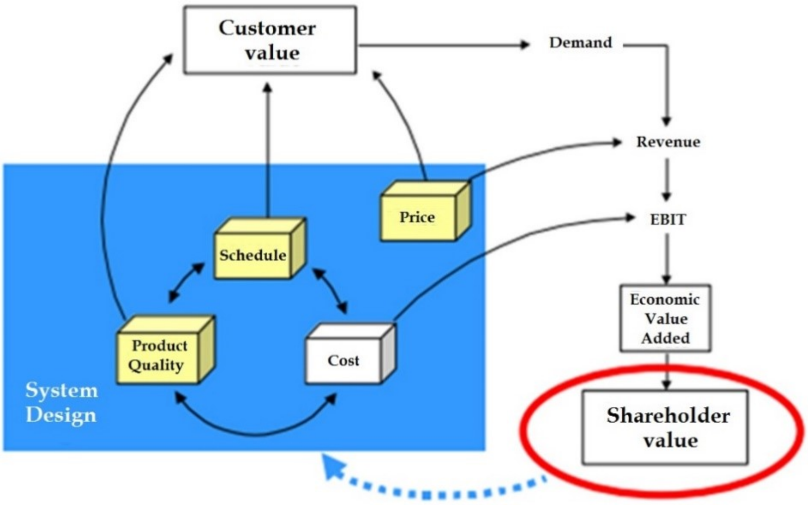


Figure 75 Value in System Design. (Markish cited in De Weck & Willcox, 2010)

The definition of value will vary depending on the system and the role of the stakeholders; however, a quantifiable metric must be defined. Thus, the spectrum of traditional metrics should be broadened:

Traditional Metrics	Augmented Metrics
Performance	Cost
Weight	Revenue
Speed	Profit
	Quietness
	Emissions
	Commonality
	…

[5] A *shareholder* is a person who owns shares in a company and therefore gets part of the company's profits and the right to vote on how the company is controlled.

Then, designing for *value* is crucial, it is not enough to focus exclusively on performance to be successful. The definition of *value* is flexible, and will vary depending of the interest of the stakeholders. Sometimes financial metrics can be used to quantify value, but they must be used with caution (De Weck & Willcox, 2010).

Turner & Zolin (2012) propose "The new model of project success". The information shown in Table 50 can be seen as how different stakeholder types see *value* in the system and/or project through time. For example, the *value* for the investor or owner at the end of the project is: time, cost, feature, and performance; after several months his idea of *value* for the same project has changed to: performance, profit, reputation, and consumer loyalty.

Table 50 New model of project success (adapted from Turner & Zolin, 2012)

Stakeholder types	Value at the end of the project	Value after several months of the project
Investor or owner	Time Cost Features Performance	Performance Profit Reputation Consumer loyalty
Project executive or Project sponsor	Features Performance Time Cost	Performance Benefits Reputation Relationships Invertor loyalty
Consumers	Time Price or benefit Features	Benefit Price of product Features Developments
Operators or users	Features Performance Documentation Training	Usability Convenience Availability Reliability Maintainability
Project Manager and Project team	Time Cost Performance Learning Camaraderie Retention Well-being	Reputation Relationships Repeat business
Senior supplier	Completed work Time and cost Performance Profit from work Safety record Risk record Client appreciation	Performance Reputation Relationships Repeat business
Other suppliers (goods, materials, Works, or services)	Time Profit Client appreciation	Reputation Relationships Repeat business
Public	Environmental impact	Environmental impact Social costs Social benefits

Creating value without waste

The lean process strategy seeks to reduce waste and increase flow, and thus does more in less time, with fewer resources and activities (Gutierrez Pulido, 2010). It creates value without waste.

According to Walton (1999), lean in product definition is the phase where stakeholder needs are identified and applied to a product to satisfy that need. For this author, the product definition phase includes the requirement generation and resource prioritization. The author suggests that effective requirements generation can be attributed to three things:

1. Well trained competent people;
2. Structured, tailorable process; and
3. Management support

Oehmen (2012) mentions that the process of *creating value without waste* is well defined by Six Lean Principles:

1. Capture the value defined by the key customer stakeholders
2. Map the value stream and eliminate waste
3. Flow the work through planned and streamlined processes
4. Let customers stakeholders pull value
5. Pursue perfection in all processes
6. Respect the people in your program

According to Darwish et al. (2016), there are five manufacturing lean principles that can be seen as lean product development principles. Table 51 illustrates the lean principles in the first column, the manufacturing lean principles in the second column, and the product development lean principles in the third and last column. For instance, the lean principle *identifies value stream*, in manufacturing, is seen as *parts and materials*, while, in product development, it is seen as *information and knowledge*.

Table 51 Manufacturing lean principles versus product development lean principles (Darwish et al., 2016)

Five Lean Principle	Manufacturing	Product Development
Value	Visible at each step, defined goal	Harder to see, emergent goals
Identify the value stream	Parts and material	Information and knowledge
Make the value flow	Iterations are waste	Planned iterations must be efficient
Let the customer pull the process	Driven by TAKT time[6]	Driven by the needs of enterprise
Pursue perfection	Process repeatable without errors	Process enables enterprise improvement

According to Bauch (2004), "*the product of product development is information*", "*where new and useful information is created for a new product*". This information consists of product, project, process, and roles knowledge, representing a "*very basic prerequisite for a successful development process*". Consequently, companies face one big challenge: taking care of quality information through the product development process, while standing up uncertainty and risk (Bauch, 2004).

In this perspective, quality management systems can help managing information generated by product development processes.

5.3 Lean Thinking and Quality Assurance

Besides product development processes, lean philosophy has also influenced quality assurance. Some authors like Blecken, Zobel, & Maurantzas (2010) Chiarini (2011), Adina-Petruța & Roxana (2014), Bacoup et al. (2015, 2018) and Marques et al. (2016) have studied how to relate *lean* with *quality management systems*, specifically to ISO 9001 (2015), showing that "*the*

[6] *TAKT time* is the cadence or rhythm with which an article must be produced to meet the demand.
Takt time = time available to produce / number of units consumed or sold

requirements of a quality management system are not necessarily conflicting with the lean principles" (Blecken, Zobel, & Maurantzas, 2010). Nevertheless, this topic will not be addressed in this research work.

In contrast, this section discusses if a quality management system can, or cannot, become a *lean* quality management system. In this context, only one reference was found in literature: "From a Quality Management System (QMS) to a Lean Quality Management System (LQMS)" (Bacoup et al., 2018).

The objective of these authors' research is to develop a methodology that leads companies to certification without creating more documentation than needed. The methodology was named "*Lean Normalization*", and leads to "*an integrated and agile QMS called Lean QMS*" (Bacoup et al. 2018).

The methodology is illustrated in Table 52 and contains six steps: documentary muda, right documents, design of the continuous improvement process, due quality, visual communication, and lean quality management system animation. In the first column of the table is written the lean management concept that supports the step, and in the third column is expressed the objective of every step of the "*Normalization Methodology*".

Table 52 Lean Normalization Methodology (Bacoup et al., 2018)

Lean Management concept	Step	Objective
Elimination of waste	A01 – Documentary muda	To identify the level of documentation required, the exact number
Just in time	A02 – Right documents	To match the ISO requirements with the existing documentation
Continuous improvement	A03 – Design of the continuous improvement process	To describe the processes of involving executive committee and scorecards to control the improvement process
Perfect quality	A04 – Due Quality	To apply all the basic principles to improve quality in the organization, products, and services
Visual management	A05 – Visual communication	To allow communication of the LQMS to all employees using a visual scorecard
Human resources management	A06 – Lean Quality Management System (LQMS) Animation	To train internal human resources and make all the collaborators aware of the QMS

5.4 Lean Thinking and Systems Engineering

Oehmen (2012), in his book The Guide to Lean Enablers for Managing Engineering Programs, establishes 329 "*best practices*" or "*lean enablers*", which integrate the Six Lean Principles into engineering programs. He mentions that it has already been proven that if the suggested program is followed, it will be successful in any engineering program. The author suggests applying all Lean Enablers for System Life-Cycle Phases during the Conceptual / Preliminary Design phase.

As mentioned above, according to Oehmen (2012), the challenge that ranks second in importance during management engineering programs is *Theme 2: Unstable, Unclear and Incomplete Requirements*. The Lean Enablers specifically recommended for this point are shown in

Table 53; lean enablers are listed according to groups and sub-groups; for example, lean enabler group 2.1 *Establish the value and benefit of the program to the stakeholders*, contains the sub-group *2.1.1,* for which the recommended lean enabler is: "*Define value as the outcome of an activity that satisfies at least three conditions. A. The external customer stakeholders are willing to pay for value. B. Transforms information or material or reduces uncertainty. C. Provides specified programs benefits right at the first time.*"

Table 53 Lean Enablers Directly Addressing Unstable, Unclear and Incomplete Requirements (Oehmen 2012, pp 146-148)

Lean Enabler #	Lean Enablers Addressing Challenge 2: Unclear Requirements
2.1	Establish the value and benefit of the program to the stakeholders
2.1.1	Define value as the outcome of an activity that satisfies at least three conditions. A. The external customer stakeholders are willing to pay for value. B. Transforms information or material or reduces uncertainty. C. Provides specified programs benefits right at the first time
2.1.2	Define value – added in terms of the value of the customer stakeholders and their need
2.1.3	Develop a robust process to capture, develop, and disseminate customer stakeholder value with extreme clarity
2.1.4	Proactively resolve potential conflict on stakeholder values and expectations, and seek consensus
2.1.5	Explain customer stakeholder culture to Program employees, i.e. the value system, approach, attitude, expectations, and issues
2.2	Focus all program activities on the benefits that the program intends to deliver
2.2.1	All program activities, including communications and metrics, must be focused on the intended outcomes of the program-the program's planned benefits
2.2.3	Ensure program staff and teams fully understand how program execution and benefits relate to high-level organizational goals (e.g., competitiveness and profitability)
2.3	Frequently engage the stakeholders throughout the program life cycle
2.3.1	Everyone involved in the program must have a customer-first spirit, focusing on the clearly defined program value and requirements
2.3.2	Establish frequent and effective interaction with internal and external stakeholders
2.3.3	Pursue a program vision and architecture that captures customer stakeholder requirements clearly and can be adaptive to changes
2.3.4	Establish a plan that delineates the artifacts and interactions that provide the best means for drawing out stakeholder requirements
2.3.5	Structure communication among stakeholders (who, how often, and what)
2.3.6	Create a shared understanding of program content, goals, status, and challenges among key stakeholders
2.3.7	Communicate accomplishments and major obstacles with stakeholders regularly and with transparency
2.3.8	Build trust and healthy relationships with stakeholders by establishing open communication and early engagement with the program planning and execution
2.3.9	Listen to the stakeholders' comments and concerns patiently and value their views and inputs
2.3.10	Clearly track assumptions and environmental conditions that influence stakeholder requirements and their perception of program benefits
2.3.11	Use program component selection and review with the key stakeholders as an opportunity to continuously focus the program on benefits delivery
2.4	Develop high-quality program requirements among customer stakeholders before the bidding and execution process begins
2.4.1	Ensure that the customer-level requirements defined in the request for proposal (RFP) or contracts are truly representative of the need, stable, complete, crystal clear, de-conflicted, free of wasteful specifications, and as simple as possible
2.4.2	Use only highly experienced people and expert institutions to write program requirements, RPFs, and contracts
2.4.3	If the customer lacks the expertise to develop clear requirements, issue a contract to a proxy organization with towering experience and expertise to sort out and mature the requirements and specifications in the RFP. This proxy must remain accountable for the quality of the requirements, including personal accountability
2.4.4	Prevent careless insertion of mutually competing and conflicting requirements, excessive number of requirements, standards, and rules to be followed in the program, mindless "cut-and-paste" of requirements from previous programs
2.4.5	Minimize the total number of requirements. Include only those that are needed to create value to the customer stakeholders
2.4.6	Insist that a single person is in charge of the entire program requirements to assure consistency and efficiency throughout

2.4.7	Require personal and institutional accountability of the reviewers of requirements until program success is demonstrated
2.4.8	Always clearly link requirements to specific customer stakeholders needs and trace requirements from this top level to bottom level
2.4.9	Use peer-review requirements among stakeholders to ensure consensus validity and absence of conflicts
2.4.10	Require an independent mandatory review of the program requirements, the concept of operation, and other relevant specifications of value for clarity, lack of ambiguity, lack of conflicts, stability, completeness, and general readiness for contracting and effective program execution
2.4.11	Clearly articulate the top-level objectives, value, program benefits, and functional requirements before formal requirements or a request for proposal is issued
2.4.12	Use a clear decision gate that reviews the maturity of requirements, the trade-offs between top-level objectives, and the level of remaining requirements risks before detailed formal requirements or a request for proposal is issued
2.5	Clarify, derive and prioritize requirements early, often and proactively
2.5.1	Develop and agile process to anticipate, accommodate, and communicate changing customer requirements
2.5.2	Follow up written requirements with verbal clarification of context and expectations to ensure mutual understanding and agreement. Keep the record in writing, share the discussed items, and do not allow requirements creep
2.5.3	Use architectural methods and modeling to create a standard program system representation (3D integrated CAE toolset, mockups, prototypes, models, simulations, and software design tools) that allow interactions with customers and other stakeholders as the best means of drawing out requirements
2.5.4	Listen for and capture unspoken customer requirements
2.5.5	To align stakeholders, identify a small number of primary goals and objectives that represent the program mission, how it will achieve its benefits, and what the success criteria will be to align stakeholders. Repeat these goals and objectives consistently and often
2.5.6	Actively promote the maturation of Stakeholder Requirements, e.g., by providing detailed trade-off studies, feasibility studies and virtual prototypes
2.5.7	Facilitate communication between different and possibly diverging stakeholders to develop a shared understanding of the program among the stakeholders, clearly identifying and incorporating the various interest of different stakeholders (aligned, indifferent, or opposed), and establish trust
2.5.8	Create effective channels for clarification of requirements (e.g., involving customer stakeholders in program teams)
2.5.9	Fail early and fail often through rapid learning techniques (e.g., prototyping, test, simulations, digital models or spiral development)
2.5.10	Employ agile methods to manage necessary requirements change and make the program deliverables robust against those changes. Make both program processes and program deliverables reusable, reconfigurable, and scalable
3.5.14	The program manager must personally understand, clarify and remove ambiguity, conflicts, and waste from key requirements and expectations al the program start
3.7.6	When defining requirements sets for multiple suppliers, ensure that they are independent of each other, in order to minimize risk and reduce the need to manage dependencies among suppliers
3.7.7	Communicate to suppliers with crystal clarity all expectations, including the context and need, and all procedures and expectations for acceptance tests, and ensure the requirements are stable
3.10.8	Match technologies to program requirements. Do not exceed program needs by using unnecessarily exquisite technologies ("gold plating")
3.10.9	Perform robust system architecting and requirement analysis to determine technology needs and current technology readiness levels
4.9.2	Be willing to challenge the customer's assumptions on technical and mediocratic grounds, and to maximize program stability, relying on technical expertise
4.10.7	Align program metrics with intended benefits and stakeholder expectations
5.1.6	For non-routine tasks, avoid rework by coordinating task requirements with internal customer
6.1	Make effective use of existing program management and organizational maturity standards

As mentioned before in Chapter 1, the processes under study belong to the Systems Engineering Technical Processes; in this context, the processes are 4.1 Stakeholder Needs and Requirements Definition Process, and 4.2 System Requirements Definition Process. For these processes, Oehmen (2012) specifically recommends the application of the following Lean Enablers shown in Table 54. Table 54 illustrates the lean enablers according to the Systems Engineering (SE) process number, and the lean enabler number.

Table 54 Lean Enablers Sorted by Systems Engineering Process Number (Oehmen 2012, pp. 177-178)

SE Process #	Lean Enabler #	Lean Enabler for Managing Engineering Programs
4		**Systems Engineering: Technical Processes**
4.1		**Stakeholder Requirements Definition Process**
4.1	2.1	Establish the value and benefit of the program to the stakeholders
4.1	2.1.1	Define value as the outcome of an activity that satisfies at least three conditions. A. The external customer stakeholders are willing to pay for value. B. Transforms information or material or reduces uncertainty. C. Provides specified programs benefits right at the first time
4.1	2.1.2	Define value – added in terms of the value of the customer stakeholders and their needs
4.1	2.1.3	Develop a robust process to capture, develop, and disseminate customer stakeholder value with extreme clarity
4.1	2.1.4	Proactively resolve potential conflicting stakeholder values and expectations, and seek consensus
4.1	2.1.5	Explain customer stakeholder culture to Program employees, i.e. the value system, approach, attitude, expectations, and issues
4.1	2.3.10	Clearly track assumptions and environmental conditions that influence stakeholder requirements and their perception of program benefits
4.1	2.5	Clarify, derive and prioritize requirements early, often and proactively
4.1	2.5.4	Listen for and capture unspoken customer requirements
4.1	2.5.6	Actively promote the maturation of Stakeholder Requirements, e.g., by providing detailed trade-off studies, feasibility studies and virtual prototypes
4.1	2.5.7	Facilitate communication between different and possibly diverging stakeholders to develop a shared understanding of the program among the stakeholders, clearly identifying and incorporating the various interest of different stakeholders (aligned, indifferent, or opposed), and establish trust
4.1	2.5.8	Create effective channels for clarification of requirements (e.g., involving customer stakeholders in program teams)
4.1	2.5.10	Employ agile methods to manage necessary requirements change and make the program deliverables robust against those changes. Make both program processes and program deliverables reusable, reconfigurable, and scalable
4.1	3.5.14	The program manager must personally understand, clarify and remove ambiguity, conflicts, and waste from key requirements and expectations at the program start
4.2		**Requirements Analysis Process**
4.2	3.10.9	Perform robust system architecting and requirement analysis to determine technology needs and current technology readiness levels

The methodology resulting from this research work should include, as a minimum, the Lean Enablers of Table 54 proposed by Oehmen (2012).

5.5 Conclusion

This chapter reviewed Lean Thinking philosophy. The three fundamental concepts of *value*, *waste*, and *creating value without waste* were explained and directly applied to product development processes. Value-stream-map was introduced as a useful tool to identify *waste* in processes; as well, different types of waste for product development processes were presented. The term *value* was interpreted in the context that there are activities that add value to processes, and certain others that do not; additionally it was explained that, from the point of view of a product, the *value* differs from one to another stakeholder in the same project.

In addition, this chapter explained how Lean Thinking philosophy has influenced Quality Assurance and Systems Engineering domains. It was shown how quality management systems have been positively affected by *lean* regarding documentation for ISO certifications purposes. For *lean* influence in Systems Engineering, there were presented the lean enablers for addressing the question of unclear requirements, as well as the lean enablers for managing engineering programs related to the processes Stakeholder Requirements Definition Process, and Requirements Analysis Process.

Applying correctly lean concepts will lead us to achieve the goal of proposing a methodology and its tools (documents in the form of a quality management system) that only includes valued-added activities and required non-value-added activities. At the same time, it is essential to remember the seven types of waste that must be avoided when re-designing the processes under study. In addition, the lean enablers for managing engineering programs (Oehmen, 2012) will be considered to reduce risks of project abortion or delays.

Next chapter will develop our proposal, integrating several author's contributions.

CHAPTER 6 DREAM: a Quality Assurance Methodology for System Requirements definition

6.1 Introduction

"Researchers have made considerable advancement in the area of requirements engineering but still development is needed… further work is required such as requirements preprocessing, risk management, requirements prioritization, application of a specific elicitation technique, etc." (Batra & Bhatnagar, 2017).

This chapter presents *DREAM* (Driven Requirements Analysis Management), the methodology resulting from this research work. It is developed as two main processes: a) *from stakeholder needs to stakeholder requirements*, and b) *from stakeholder requirements to system requirements*.

Both processes were developed taking in consideration different points of views found in the literature. In that sense, the processes we propose to integrate coherently several contributions in order to be strong processes able to assure the quality system while reducing waste.

In some activities (or tasks), we had to decide between different options of how to conduct them; in this respect, we analyzed the possibilities considering the available information resulting from the previous activity, the objective of the current activity, the expected information needed to continue with the following activity, and how we can add value while reducing waste.

In other cases, there was not a clear path or clue to follow; for example, there is not an objective method to *weigh* the stakeholder level of importance, nor any structured method that helps to elicit needs, nor even a clear definition of *bi-directional traceability at adjacent levels of the system*. Nevertheless, a weakness is always an opportunity; the fact that there was no track to carry out these activities allowed us to give free rein to our creativity to make interesting proposals.

We propose that both processes are linear with the possibility of iterations as needed; the activities' order was decided after reflecting consequences if following one or another order possibility, trying at all times to save the maximum of available resources and avoid re-work. It is, *do it right at the very first time*.

The method followed for the conception of DREAM is illustrated in Figure 76: literature review was first conducted to identify the main activities of both processes (a) and to precise how each activity was executed (b); then, the different authors' point of views were analyzed to identify the differences in how to perform the activities and in what order (c); at this point, the methodology was designed through making the decisions of: what activities to include, in what order, what tasks are necessary to perform every activity, and integrating several author's contributions to design each task, or develop creative task design (d); DREAM was then verified through its compliance with ISO 15288 (2015) (e), and finally, DREAM was validated with several case studies (f).

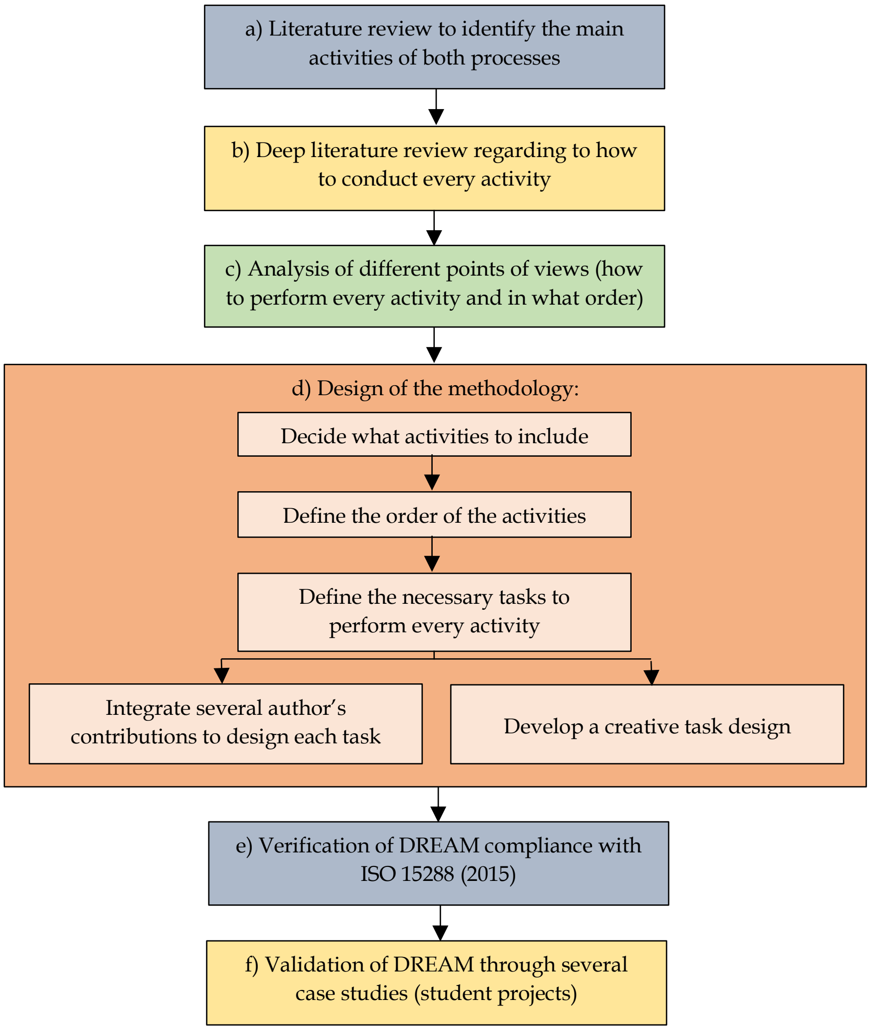


Figure 76 Research methodology to design the DREAM methodology

In the next sections the reader will find the main processes, the description of the activities included in each process, and the necessary tasks to perform every activity. In addition, Chapter 6 presents the proposed supporting tools designed as a quality management system named *DREAM as a QMS*. See Annex 3.

DREAM as a QMS was designed with the objective to provide support for non-expert users when developing new systems. It consists of the following documentation: two processes (P), eleven work instructions (W) and twenty forms (F):

- **Processes:** *"set of interrelated or interacting activities that transforms inputs into outputs"* (ISO 24765, 2017, p. 337)
- **Work Instructions** (W) or **Procedures**: *"information item that presents an ordered series of steps to perform a process, activity, or task"* (ISO 24765, 2017, p. 336)
- **Forms** (F): *"module or formulary to collect data"* (ISO 24765, 2017, p. 187)

Once the forms are applied in the design of a system, they will become auditable **records**: *"set of related data items treated as a unit"* (ISO 24765, 2017, p. 363).

Chapter 7 will present the last steps (*e* and *f*) about DREAM verification and validation.

Based on literature (Brahm & Kleiner, 1996; Hull, Jackson, & Dick, 2011) we suggest that DREAM methodology is conducted within a collaborative environment, where the Nominal Group Technique (NGT) can allow to a highly effective group decision making process.

6.2 From Stakeholder Needs to Stakeholder Requirements process

Figure 77 illustrates the proposed process to transform *stakeholder needs into stakeholder requirements*; on the left side are shown the process activities and on the right side the necessary tasks to perform the activity. This process is iterative; it means that at any time it could be done again and again in order to improve the results.

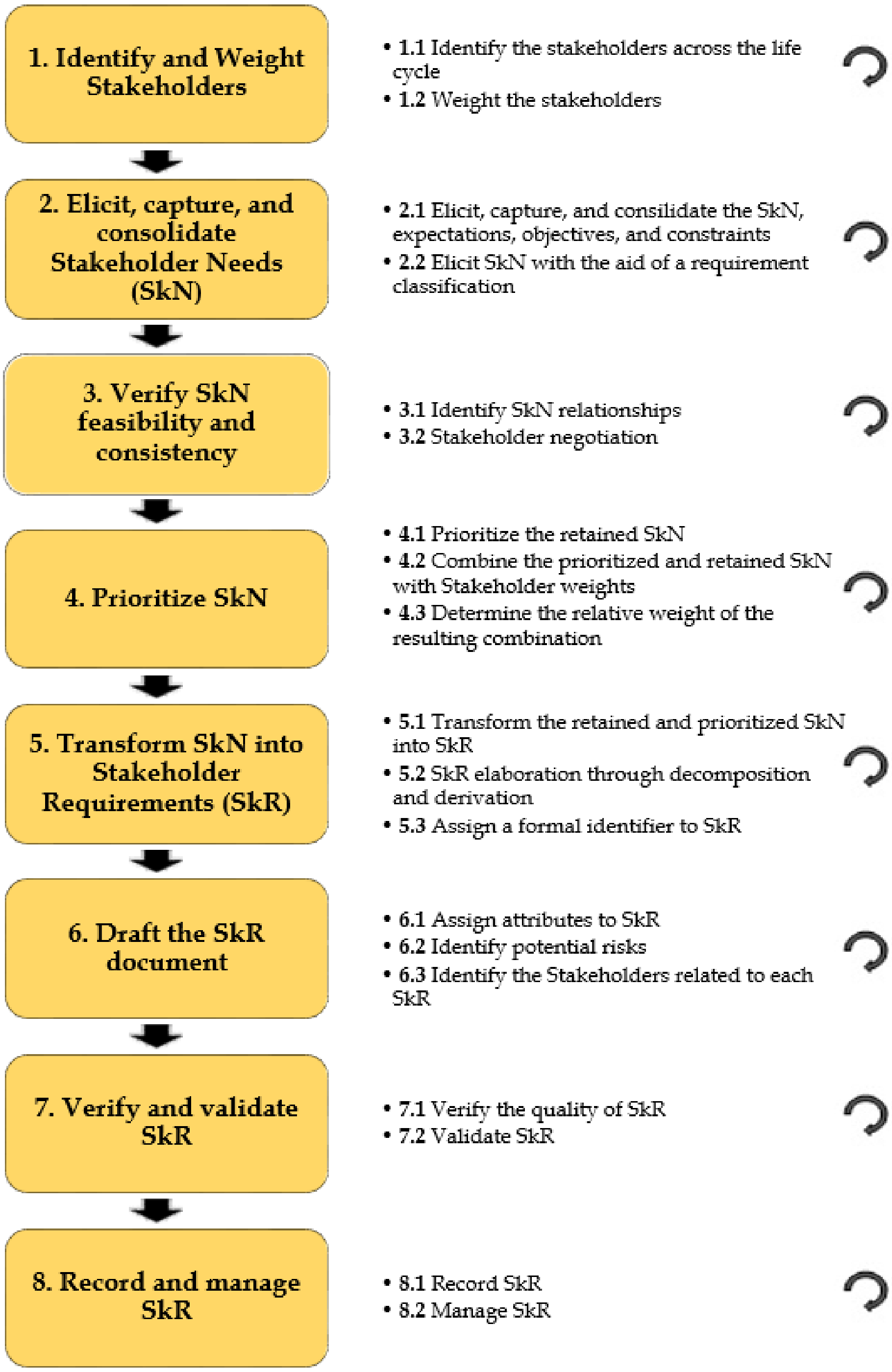


Figure 77 Proposal: stakeholders needs to stakeholders requirements process

NOTE: This first part of the methodology was presented at the 3rd National Multidisciplinary Congress of Education, Science, and Technology CONAMTEC, on November 2017, Hidalgo, Mexico; later, the article "Metodología para transformar necesidades en requisitos, integrando Ingeniería de Sistemas, Calidad y Pensamiento Esbelto" was published in the Journal ECORFAN Revista de Aplicaciones de la Ingeniería. Vol. 1, No. 2, pp. 42 – 51.

(The article has open access through the link: http://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ingenieria_Industrial/vol1num2/ECORFAN_Revista_de_Ingenier%C3%ADa_Industrial_V1_N2_4.pdf)

Table 55 shows the relationship among the activities and task of the proposed process to transform *stakeholder needs into stakeholder requirements*, related to the work instructions and forms of DREAM as a QMS. See Annex 3.

Table 55 Relationship among methodology and DREAM as a QMS

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
1. Identify and Weight Stakeholders	1.1 Identify the Stakeholders across the life cycle	W-1 Identify and Weight the Stakeholders	F-1a Identify and Weight the Stakeholders
	1.2 Weight the Stakeholders		F-1b List of Stakeholders
2. Elicit, capture, and consolidate Stakeholder Needs (SkN)	2.1 Elicit, capture, and consilidate the SkN, expectations, objectives, and constraints	W-2 Define system mission, purpose and objectives	F-2a Define system mission, purpose and objectives
			F-2b Collaborative session to define system mission, purpose and objectives
			F-2c System mission, purpose and objectives
	2.2 Elicit SkN with the aid of requirement classification	W-3 Elicit Stakeholder Needs	F-3a Elicit Stakeholder Needs
			F-3b Collaborative Session Elicit Stakeholder Needs
			F-3c Stakeholder Needs
3. Verify SkN feasibility and consistency	3.1 Identify SkN relationships	W-4 Stakeholder Needs feasibility and consistency	F-4a Stakeholder Needs feasibility and consistency
	3.2 Stakeholder negociation		F-4b Stakeholder Negotiation
4. Prioritize SkN	4.1 Prioritize the retained SkN	W-5 Prioritize Stakeholder Needs	F-5 Prioritize Stakeholder Needs
	4.2 Combine the prioritized and retained SkN with Stakeholder weights		
	4.3 Determine the relative weight of the resulting combination		
5. Transform SkN into Stakeholder Requirements (SkR)	5.1 Transform the retained and prioritized SkN into SkR	W-6 List of Stakeholder Requirements	F-6 List of Stakeholder Requirements
	5.2 SkR elaboration through decomposition and derivation		
	5.3 Assign a formal identifier to SkR		
6. Draft the SkR document	6.1 Assign attributes to SkR	W-7 Stakeholder Requirements	F-7a Stakeholder Requirements
	6.2 Identify potential risks		
	6.3 Identify the stakeholders related to each SkR		
7. Verify and validate SkR	7.1 Verify the quality of SkR		F-7b Set of Stakeholder Requirements
	7.2 Validate SkR		
8. Record and manage SkR	8.1 Record SkR		
	8.2 Manage SkR evolution or modification		

In the following subsections it will be explained for each activity and its tasks:

- *Our proposal*: this section shows the bibliographical support and our proposition
- *How to do it*: this section explains how to develop the task.
- The *support documents:* work instructions (W) and forms (F) that are part of DREAM as a QMS.

Activity 1: Identify and Weight the Stakeholders

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
1. Identify and Weight Stakeholders	1.1 Identify the Stakeholders across the life cycle	W-1 Identify and Weight the Stakeholders	F-1a Identify and Weight the Stakeholders
	1.2 Weight the Stakeholders		F-1b List of Stakeholders

Task 1.1: Identify the Stakeholders across the lifecycle

It is extremely important that the involved stakeholders be identified through the system life cycle, they should be taken into account and participate actively (Freeman & Reed, 1983) since de *Concept* stage of the system life cycle.

Our Proposal:

To integrate the contributions of SEBoK (2016) related to system life cycle and their purpose; the contributions of Turner & Zolin (2012) related to 8 stakeholders types; the contributions of Faisandier (2012) related to take into account all the stakeholders through system life cycle, and completion of stakeholders types (*people who don't want the system or maliciously intended*); Sharp, Finkelstein & Galal (1999) to complete the stakeholders types (*legislators*). To include the *gemba* visit proposed by Mazur (2012), Bylund, Wolf, & Mazur (2009), Alshenqeeti (2014), and Schillo, Isabelle, & Shakiba (2017) to recognize the involved stakeholders.

This way, we propose 10 stakeholder types:

1. **Legislators:** professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for operation that will affect the development and/or operation of the system.
2. **Owner or investor:** who pay, buy, pay for its operation, and obtain revenue
3. **Project executive or project sponsor:** prior to the Project, identify the need for a new asset and the potential benefit it will bring
4. **Consumers:** who buy the product the new assets produces
5. **Operators/users:** who operate the asset on behalf of the owner
6. **Project manager and project team:** who manage the project, the analysis and design team in the requirements engineering process
7. **Senior supplier** (design and/or management): senior management in the lead contractor
8. **Other suppliers:** people or groups who provide goods, materials; Works or services
9. **Public:** the public concerned with the environmental and social impacts of the system, they will want to know how their taxes have been spent
10. **People who don't want the system or maliciously intended:** to identify possible risks or threats

How to do it:

We suggest that the Project manager and the analysis and design team apply the technique of NGT (Brahm & Kleiner, 1996) to identify the involved stakeholders in the system life cycle; this would avoid the common mistakes during the system requirement definition process described by Faisandier (2012). And finally, to manage the project as stated by the lean enablers described by Oehmen (2012):

- *"Engage the stakeholders throughout the program life cycle.*
- *Establish frequent and effective interaction with internal and external stakeholders.*
- *Build trust and healthy relationships with stakeholders by establishing open communication and early engagement with program planning and execution.*
- *Facilitate communication between different and possibly diverging stakeholders to develop a shared understanding of the program among the stakeholders, clearly identifying and incorporating the various interest of different stakeholders (aligned, indifferent, or opposed), and establish trust."*

Task 1.2: Weight the Stakeholders

Our Proposal:

Knowing that:

a) Glinz & Wieringa (2007) highlight that the first activity of Requirements Engineering must be identified *who* the stakeholders are, and determine *how important* they are;
b) Mazur (2012) and Jepsen & Eskerod (2009) express that it is necessary to consider the *importance* of the stakeholders because depending on it, his or her needs will be taken in major or minor degree;
c) Narendhar & Anuradha (2017) state that the first key activity of the Win-Win negotiation model is to identify the success-critical stakeholders who will negotiate and make decisions if needed; and
d) Freeman & Reed (1983) recommend assessing the *relative stakeholder importance*;

We propose that once the stakeholders are identified through the system life cycle, the stakeholders be *weighted* in accordance with the attributes proposed by Mitchel, Agle, & Wood (1997):

> **Power:** a relationship among social actors in which one social actor, A, can get another social actor, B, to do something that B would not have otherwise done
>
> **Legitimacy:** a generalized perception or assumptions that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values, beliefs, definitions
>
> **Urgency:** the degree to which stakeholder claims call for immediate attention

Mitchel, Agle, & Wood (1997) define *salience* as: "the degree to which managers give priority to competing stakeholder claims; the authors state that "power gains authority through legitimacy, and it gains exercise through urgency".

The stakeholder weights will be helpful mostly in two future tasks:

a) Prioritize needs/ requirements, and find where the *value* is
b) Take major decisions when negotiating

Considering these facts and, with the objective of eliminating the subjectivity in the process of weight the stakeholders, we propose to assign numerical weights to the stakeholder attributes. These proposed weights were assigned arbitrarily because no precedent was found in the consulted literature regarding stakeholder numerical evaluation. We propose the following rubric, where values were assigned considering that power is more important than legitimacy, and that legitimacy is more important than urgency according to Mitchel, Agle, & Wood (1997):

Proposed weight	0.00	1.00	2.00	3.00
Power	This stakeholder cannot influence or make decisions to determine the actions to follow	Sometimes the stakeholder may influence the decisions to determine the actions to follow	Sometimes the stakeholder can make decisions and/or determine the actions to follow	The power is absolute, this stakeholder can make decisions and/or determine the actions to follow
Proposed weight	0.00	0.50	1.00	1.50
Legitimacy	Stakeholders that don't have a legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have a little legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have some legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have total legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome
Proposed weight	0.00	0.25	0.50	0.75
Urgency	Managerial delay in attending to the claim or relationship is acceptable to the stakeholder AND/OR Stakeholders who don't claim for immediate attention	Managerial delay in attending to the claim or relationship is sometimes acceptable to the stakeholder AND/OR Stakeholders who few times claim for immediate attention	Managerial delay in attending to the claim or relationship is almost never acceptable to the stakeholder AND/OR Stakeholders who almost ever claim for immediate attention	Managerial delay in attending to the claim or relationship is unacceptable to the stakeholder AND/OR Stakeholders who claim for immediate attention

How to do it:

Every stakeholder should be assessed in the three attributes in relation to its position within the project; for example, if the stakeholder:

> Sometimes the stakeholder can make decisions and/or determine the actions to follow
> **Power weight = 2.00**
>
> Stakeholders that have total legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome
> **Legitimacy weight = 1.50**
>
> Managerial delay in attending to the claim or relationship is acceptable to the stakeholder AND/OR Stakeholders who don't claim for immediate attention
> **Urgency weight = 0.00**

Then, the **total stakeholder weight** will be obtained through the addition of the individual *weights* of each attribute; for the example, the **total stakeholder weight** is:

2.00 + 1.5 + 0.00 = **3.50**

It is possible that a stakeholder appears several times in the system life cycle and it is possible that the *weights* obtained each time are different; in this case, it is suggested to take into account the greater total stakeholder weight as her/his value. For example:

Concept stage	**total stakeholder weight** = 3.25
Production stage	**total stakeholder weight** = 4.00
Retirement stage	**total stakeholder weight** = 3.50

The total stakeholder weight to consider is the *maximum* obtained, in our example is 4.00
It is possible that several stakeholders have the same *maximum,* it is normal, it means that these stakeholders are equally important for the project.

NOTE: When the total stakeholder weight is equal to zero, this person or organization is a non-stakeholder, she/he/it has to be removed from the stakeholder group.

This *maximum* will be used a posteriori in the Activity 4: Prioritize Stakeholder Needs, Task 4.3: Determine the relative stakeholder need value of the resulting combination.

Finally, we **propose** to manage the project as stated by the lean enablers described by Oehmen (2012):

- *"Create a shared understanding of program content, goals, status, and challenges among key stakeholders*
- *Communicate accomplishments and major obstacles with stakeholders regularly and with transparency*
- *Use program component selection and review with the key stakeholders as an opportunity to continuously focus the program on benefits delivery."*

NOTE: Our proposal to conduct this activity was presented at the 12e Conférence Internationale de Modélisation, Optimisation et Simulation. MOSIM 2018, on June 2018, Toulouse, France, with the article entitled "Avez-vous identifie toutes les parties prenantes?" (https://hal.archives-ouvertes.fr/hal-01989427/file/mosim2018.pdf, pages 117 to 124.)

Activity 2: Elicit, capture, and consolidate Stakeholder Needs

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
2. Elicit, capture, and consolidate Stakeholder Needs (SkN)	**2.1 Elicit, capture, and consilidate the SkN, expectations, objectives, and constraints**	**W-2 Define system mission, purpose and objectives**	**F-2a Define system mission, purpose and objectives**
			F-2b Collaborative session to define system mission, purpose and objectives
			F-2c System mission, purpose and objectives
	2.2 Elicit SkN with the aid of requirement classification	**W-3 Elicit Stakeholder Needs**	**F-3a Elicit Stakeholder Needs**
			F-3b Collaborative Session Elicit Stakeholder Needs
			F-3c Stakeholder Needs

Task 2.1: Elicit, capture, and consolidate the Stakeholder Needs, expectations, objectives, and constraints

Once stakeholders have been identified and weighted, it is necessary to define the system mission, purpose, and objectives.

Our Proposal:

Our proposal is based on the suggestions made by Sadig & Sahraoui (2016), Faisandier (2012, 2014), Mazur (2012), De Weck, Eckert, & Clarkson (2007), and Oehmen (2012). These authors suggest starting the analysis by the problem definition, continuing with the purpose of the system, its mission, and objectives, then the possible operational and technical scenarios, the system interactions with other systems, objects, stakeholders. In addition, they suggest defining what is considered as *value* for the clients and *success* for the project (Mazur, 2012). It should be considered the underlying assumptions upon which the system are based (Sadig & Sahraoui, 2016), the possible opportunities for improvements to be delivered by the system, and the implementation details relating to the system; and the risk and uncertainty (De Weck, Eckert, & Clarkson, 2007).

Faisandier (2014), Ryan & Faulconbridge (2016), Mazur (2012), and Oehmen (2012) suggest an early involvement of stakeholders in this stage of the project; by their side, Sadig & Sahraoui (2016) suggest considering the fifteen core information type (project, deliverable, system, objectives, assumptions, constraints, environment, opportunities, challenges, risks, stakeholders, processes, functional, non-functional, and implementation) during the requirement elicitation stage.

How to do it:

For the stakeholder need elicitation, we propose to apply individual questionnaires to each stakeholder previous to a collaborative session. Then, in a collaborative session, the stakeholders should develop deeper all the concepts and make consensus[7] to define the system mission, purpose, and objectives; to this aim, we suggest using the nominal group technique (NGT). After the collaborative session, one document should be filled to keep the records of the consensus.

We propose the following key questions:

1. **Analysis of the contextual situation**: What is the current situation? Define the problem. What is "*success*" for this project? What does "*value*" mean for stakeholders?
2. **Operational and technical concept:** Which is the desired service? What does the system use? What does the system procure? What does the system exchange?
3. **Operational and incident scenarios**: What shall the system do to give this service? What are the operational scenarios? What are the incident scenarios?
4. **Purpose definition:** "*Why creating a new system? Why improving existing products, services, or enterprises? What should be its utility, or its usage within the context?*" (Faisandier, 2012)
5. **Mission definition:** "*What is it supposed to do, perform, transform, provide?*" (Faisandier, 2012)

[7] Consensus can be achieved through conflict resolution. According to Badreau & Boulanger (2014), some conflict resolution techniques are: agreement, compromise, vote, variant formation, "the leader always right", "consider all of them done", strengths and weakness, and decision matrix.

6. **Objectives definition:** *"How many elements could it transform, or produce? What duration to perform the transformation or production? How many times? What frequency?"* (Faisandier, 2012)
7. **Study of the context of use:** What are the different elements of the context (systems, objects, and stakeholders) in relationship with the system in the modes related to its operation?
8. **Assumptions:** What are the assumptions upon which the system is based? What are the possible opportunities for improvements to be delivered by the system? What are the implementation details relating to the system?
9. **Risks and uncertainty**: *"Will the product, system or artifact that is being designed meet its functional and form requirements once it is on sale or in use? Will it function properly and perform adequately? Are the functional and form requirements the right ones that will lead to market success?"* (De Weck, Eckert, & Clarkson, 2007). Where does uncertainty come from that could affect the future success of the system? Does uncertainty come from endogenous factors as product context and corporate context? Does uncertainty come from exogenous factors as use context, market context, political or cultural context? Can the uncertainty be resolved by simply delaying decisions and waiting until time *x*? Can the uncertainty be represented as a random variable or as a discrete future scenario? What modeling approach can be used to quantitatively capture the uncertainty?

We suggest applying the *gemba* (Mazur, 2012; Bylund & Mazur, 2009, Alshenqeeti, 2014), this way a better understanding of stakeholder needs could be done; finally, apply the lean enablers described by Oehmen (2012):

- *"Establish the value and benefit of the program to the stakeholders*
- *Listen to the stakeholders' comments and concerns patiently and value their views and inputs,*
- *Clearly articulate the top-level objectives, value, program benefits, and functional requirements before formal requirements or a request for proposal is issued*
- *To align stakeholders, identify a small number of primary goals and objectives that represent the program mission, how it will achieve its benefits, and what the success criteria will be to align stakeholders. Repeat these goals and objectives consistently and often."*
- *Define value as the outcome of an activity that satisfies at least three conditions. A. The external customer stakeholders are willing to pay for value. B. Transforms information or material or reduces uncertainty. C. Provides specified programs benefits right at the first time*
- *Define value–added in terms of the value of the customer stakeholders and their needs*
- *Explain customer stakeholder culture to program employees, e.g. the value system, approach, attitude, expectations, and issues*
- *Focus all program activities on the benefits that the program intends to deliver*
- *All program activities, including communications and metrics, must be focused on the intended outcomes of the program-the program's planned benefits*
- *Ensure program staff and teams fully understand how program execution and benefits relate to high-level organizational goals (e.g. competitiveness and profitability)*
- *Everyone involved in the program must have a customer-first spirit, focusing on the clearly defined program value and requirements*
- *Clearly track assumptions and environmental conditions that influence stakeholder requirements and their perception of program benefits*

Task 2.2: Elicit stakeholder needs with the aid of a requirement classification

Our Proposal:

Our propositions at this point are: a) the stakeholder requirement classification, and b) the system requirement classification, shown respectively in Figure 78 and Figure 79.

Our proposals are primarily based on the stakeholder and system requirement classifications stated by Faisandier (2012), considered in this research work as the completest; in addition, our proposition integrates:

a) Functional and non-functional requirements (Glinz, 2005; Badreau & Boulanger, 2014; Mabrok, Efatmaneshnik, & Ryan, 2015)
b) *Adaptability requirements* (SEBoK, 2016)
c) "*ilities*" (De Weck, Rhodes, & Ross, 2012)
d) *Retirement terms* (disposal and recovery constraints, Ryan, 2014)
e) *Forensic requirements*, *incident modes* and *incident scenarios* (Grispos et al., 2017)

The recommendation of Glinz (2005): a requirement classification based on four facets, kind, satisfaction, representation, and role, is taken into account as follows: *representation* and *satisfaction* facets will be considered in verification and validation activities, *role* facet will be considered as an attribute when managing requirements, and *kind* facet is considered inside the requirement typology proposed by Faisandier (2012). Figure 78 illustrates our proposed stakeholder requirement classification; it contains different types of stakeholder requirements as: modes and scenarios, expected services, expected effectiveness or performance, interfaces, operational conditions, constraints, forensic requirements, and validation requirements; if available, there are shown sub-categories and sub-sub-categories of requirements.

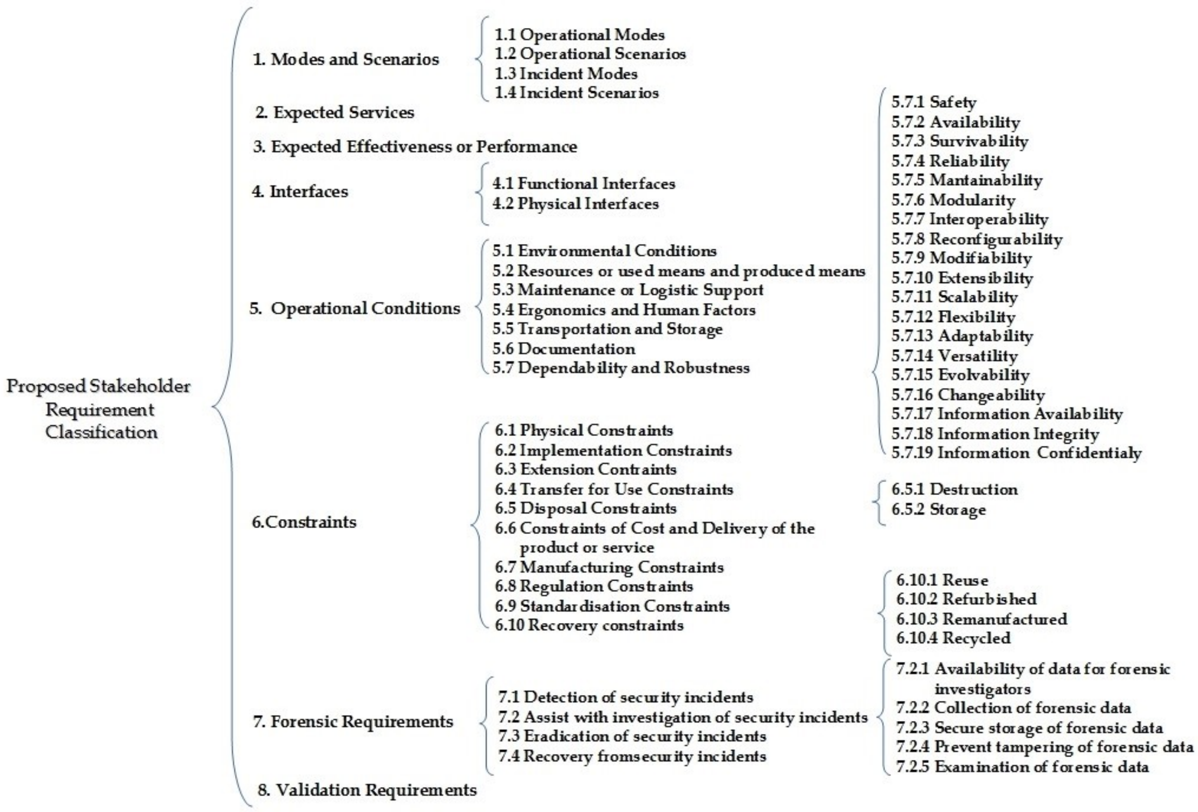


Figure 78 Proposed Stakeholder Requirement Classification

Figure 79 illustrates our proposed system requirement classification; it contains different types of system requirements as: functional requirements, non-functional requirements (effectiveness or performance requirements, interface requirements, utilization or operational requirements, forensic requirements, and validation requirements), and constraints; if available, there are shown sub-categories and sub-sub-categories of requirements.

As can be seen, stakeholder requirement classification and system requirement classification correspond to each other. Note that:

1. *Modes and Scenarios* in Figure 78, corresponds to *4.1 Modes and Scenarios* in Figure 79.
2. *Expected services* in Figure 78, corresponds to *1. Functional requirements* in Figure 79.

After proposing both requirement classifications, we propose to guide the elicitation of stakeholder needs by oriented questions; questions are based on the proposed system requirement classification in order to consider all kinds of needs during the elicitation process resulting in a complete set of requirements (Ryan et al., 2015). We suggest considering the lean enabler (Oehmen, 2012):

- *"Establish a plan that delineates the artifacts and interactions that provide the best means for drawing out stakeholder requirements."*

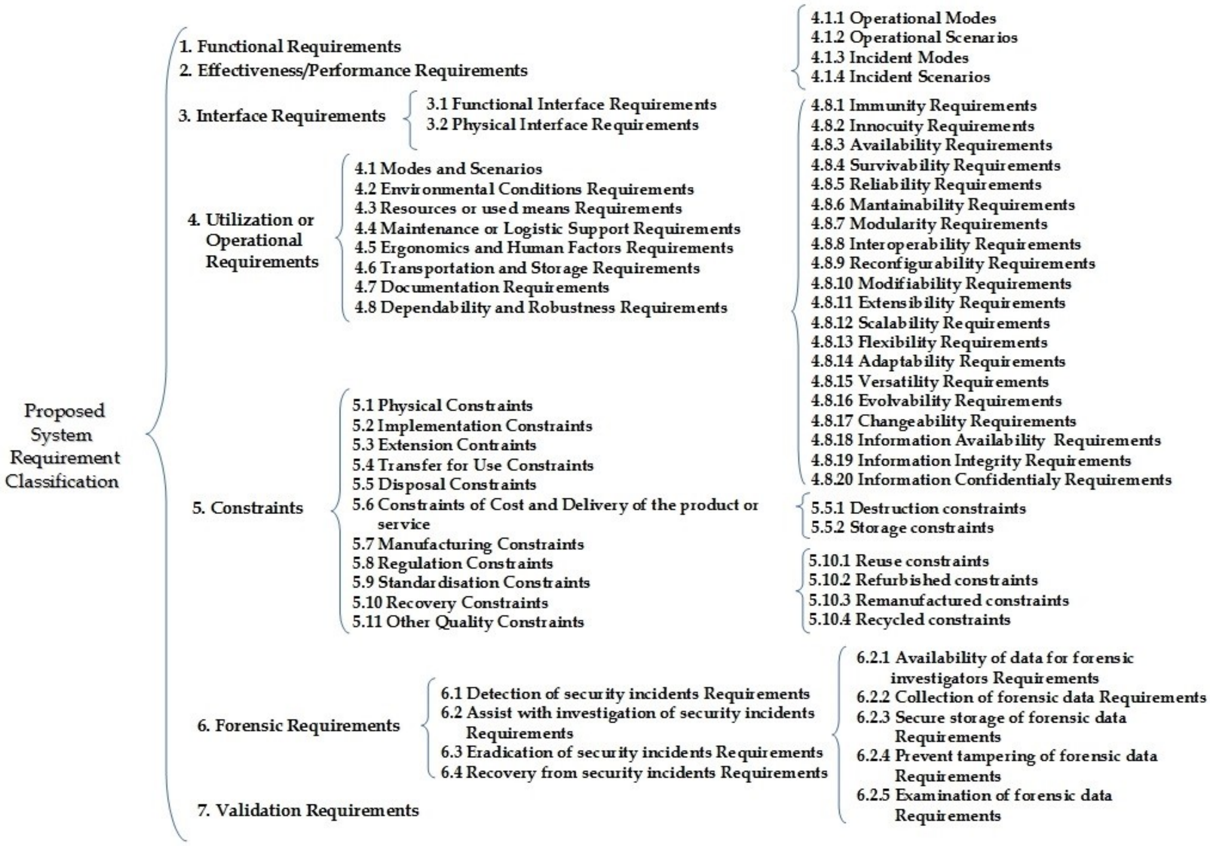


Figure 79 Proposed System Requirement Classification

How to do it:

We propose to develop the task in three steps: individual stakeholder participation, collaborative session to develop deeper all the concepts, and finally integration of consensuses stakeholder needs.

We propose the questionnaire shown in Table 56.

Table 56 Proposed questionnaire to elicit stakeholder needs

1. Functional Requirements
1. What are the main operational activities or highest level functions that the system of interest has to achieve?
2. Effectiveness / Performance Requirements
2. What is the expected performance or effectiveness of the system? What is the quality metric of effectiveness? Describe the measure of effectiveness (performances) expected by the system to satisfy the expected service (quantitative data as much as possible or at least qualitative data).
3. Interface Requirements
3.1. What are the functional interfaces between the system and the components of its operational context? Note: the human-system interfaces are described in the section of ergonomics 3.2 What are the physical interfaces that connect the system to the components of its context of use? Describe. These physical interfaces may be electrical cables, connectors, pipes, data format, protocols, procedures, etc.
4. Utilization or Operational Requirements
4.1.1 Which are the Operational Modes (on/off, standby, run, maintenance, etc.)? What is the system expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one? 4.1.2 Describe each Operational Scenario (sequence, the concurrence of actions) of the system and the exchanges with the components of the context, including the actions of the users. 4.1.3 What are the Incident Modes? What the system is expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one? 4.1.4 Describe each Incident Scenario (sequence, the concurrence of actions) of the system and the exchanges with the components of the context, including the actions of the users. 4.2 Which are the physical conditions to which the system is submitted? Describe. 4.3 What are the consumed resources or produced elements by the system, which do not belong to the system? Is it possible to define the requested autonomy, expected maximum of consumption or rejections, etc.? 4.4 Which are the requirements concerned to conditions of maintenance and logistics, duration of maintenance actions, management of spare parts, availability of maintenance equipment, qualification of the maintenance team, possibility or not for having specific tools, marking, identification, etc.? Consider the serviceability of the system: easy of inspection, cleaning, servicing, and maintenance 4.5 What are the expected man-system interfaces (control based, desk, panel's instrument, etc.)? What are the available commands and information (as well as the support of these commands and information)? For example: steering wheel, joystick, keyboard, buttons, indicator, screen, pictograms, the language used, etc. What is the priority allowing classifying the commands and the display of data? What is the appropriate user training, acceptable mental load, etc.? Describe. What are the disturbing elements resulting from the environment that could influence the operator (noise, heat, explosion, stress, etc.)? Describe. 4.6 What are the requirements concern transportation? Example: transportation conditions. What are the requirements concern storage? Example: duration and conditions for storage. What are the requirements concern handling? Example: available handling equipment 4.7 What are the requirements concerning the user documentation as installation, operating and utilization of the system, preventive maintenance procedures, training handouts, etc.? 4.8.1 Which is the expected system ability to resist natural, accidental or unintentional external threats? How the integrity of operations and performances of the system will continue even in case of failure? 4.8.2 Which is the expected system ability to guarantee the protection of the environment (people and goods) against its own actions? How will the system avoid that failures appear involving damage for people, materials or itself? 4.8.3 Which percentage of the time, ratio or average time the system will operate without failure a requested function at any time, under given environmental, maintenance and usage conditions?

4.8.4 What is the aptitude of the system to achieve its mission including the occurrence of internal failures, external threats of its environment, taken into account the expected degraded states?
4.8.5 Which is the expected ability of the system to achieve a requested function without failure, during a given period of time, under given environmental and usage conditions? Which is the expected mean time to failure (MTTF)? Which is the expected mean time between failures (MTBF)? Which are the expected failure rate per hour, km, number of cycles, etc.?
4.8.6 What are the needs for maintainability? For example: downtime for preventive maintenance actions, speed for repairing, etc. Define the objectives to be reached out concerning maintenance
4.8.7 In which degree should the system be composed of modules? Which modules are necessaries?
4.8.8 How much effectively the system should interact with other systems? What are these systems?
4.8.9 Is it necessary that the system changes its component arrangement and links reversibly? What components?
4.8.10 Is it necessary that the system changes the current set of specified system parameters? How much?
4.8.11 Is it necessary that the system accommodates new features after design? Which new features?
4.8.12 Is it possible that the current level of a specified system parameter may change? What parameters? How much?
4.8.13 Is it necessary that the system may be changed by a system-external change agent with intent? What agent(s)? When? In which circumstances?
4.8.14 Is it necessary that the system may be changed by a system-internal change agent with intent? Which agent(s)? When? In which circumstances?
4.8.15 Is necessary to satisfy diverse needs for the system without having to change form (a measure of latent value)? In what situations?
4.8.16 In what cases the system design is inherited and changed across generations (over time)?
4.8.17 When does the system needs to alter its operations or form, and consequently possibly its function, at an acceptable level of resources? In which situations?
4.8.18 What is the aptitude of the system to prevent a denial of access to a resource or information?
4.8.19 What preventive actions are taken against non-authorized modifications of the information?
4.8.20 Utilization
5. Constraints
5.1 What are the physical constraints such as size, weight, color, amount of space, of memory volume, etc.?
5.2 Is there imposed solutions; examples components to reuse, technology to be applied, material to be used or not, etc.? Which is the envisaged duration of the system life? Are there processes imposed on the development if no management/development plan is associated with the project?
5.3 Is there a constraint imposed for future evolutions? Example: access to the site, available means of lifting, duration of transfer for use, etc. Define as possible tasks and organization to install the products or services (number of people, competences, duration, training, etc.)
5.4 What are the constraints concerning the transfer for use?
5.5 Which are the constraints concerning the disposal actions when the system cannot be recovered by any means?
5.5.1 Does the system has no useful parts or hazardous materials? Should the system be destroyed, disposed of as waste or may be incinerated?
5.5.2 Does the system cannot be recovered but neither destroyed? Is storage it the best solution? Which documentation /information associated with the system as historical records, and the compliance with archival regulations must be kept? How long the system and its documents must be kept?
5.6 Is there a constraint of cost and delivery of the product? For example, a performance of effectiveness requirement could be decreased to offer a lower cost.

5.7 What are the constraints resulting from the manufacturing actions: reuse of tools/of complete production line, gripping, test outlet, etc.?
5.8 Which are the legislation, regulation applicable to the system?
5.9 What are the standard references applicable to the system?
5.10.1 Which are the possibilities to use the system on its retirement stage? Is possible that the system may be reused in another life cycle: 1) as a complete system: in its original role, or in a diminish role; 2) as separate system elements, in other words, the complete system cannot be reused as a whole, but are one or more of its elements useful?
5.10.2 Is it possible that at the end of system life it could be renovated, renewed or reconditioned? Is additional work required for continuing operating the system in its original role?
5.10.3 Would the system need significant work to be able to operate in its original role?
5.10.4 Is it possible to recover raw materials from the system in order to recycle and use it in a different form, or sold as scrap?
5.11 Which are another quality constraints to take into consideration?
6. Forensic Requirements
6.1 What may help the system to detect a security incident?
6.2 What data helps to discover why and how a security incident occurred?
6.2.1 Which is the aptitude of the system to prevent a denial of access to a resource or information to establish the cause of an incident?
6.2.2 How forensic data is acquired during investigations? How and what logging is done? How the system is examined?
6.2.3 How long and how forensic data must be kept?
6.2.4 How forensic data is prevented to be tampered?
6.2.5 How the forensic data is going to be analyzed in order to know the cause of the incident?
6.3 What requirements may help to avoid and prevent security incidents?
6.4 What requirements may help the system to recover by itself when a security incident has occurred?
7. **Validation Requirements**
7. What are the justification activities that provide elements or arguments to select the most effective solution among candidate solutions that satisfies the set of needs, expectations, and requirements of stakeholders? Describe. What is the validation strategy? What are the necessary documents for validation purpose? List them. Which are the activities or procedure to validate that the system satisfies the set of needs, expectations, and requirements? Ex: tests. What are the metrics and values to accept the system?

Finally, once stakeholders have made consensus, the stakeholder needs should be integrated into a complete list. It is very important to highlight that each stakeholder need should be recorded with its stakeholder author and rational.

Reflection: we are generating a big amount of data, if we maintain order, later it will be easier to manage it. We propose that at the time of eliciting needs, they are stored with a specific identifier following the structure:

SkN – X.Y.Z - a

Where:

SkN Stakeholder need

X	Number related to system requirement type classification group
Y	Number related to system requirement type classification sub-group (if available)
Z	Number related to system requirement type classification sub-sub-group (if available)
a	Consecutive number of stakeholder need

For example, SkN – 4.2.0 - 1 is the formal identifier of the stakeholder need (SkN) of environmental conditions (4.2.0), and is the first need of this type (1). The formal identifier will help to trace the specific stakeholder need with their future related stakeholder requirements and systems requirements.

NOTE: the questionnaire will lead to the elicitation of stakeholder needs assuring that all types of needs are considered, but not only this, the use of the proposed questionnaire and the proposed SkN identifier, that elicited stakeholder needs are already classified by type, it will not be necessary another extra activity for classifying the SkN. We add value to the process (through systematic and structured elicitation, and systematic SkN identification) while reducing waste of time (no extra activity of classifying SkN).

NOTE: Our proposal to conduct this activity was presented at the 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, on June 2018, Bergamo, Italy. Later, the article entitled "How to find non-functional requirements in system developments" was published at IFAC-PapersOnLine. Online ISSN: 2405-8963, Vol. 51, No. 11, pp. 1573 – 1578. (https://www.sciencedirect.com/science/article/pii/S240589631831396X)

Activity 3: Verify Stakeholders Needs feasibility and consistency

Methodology		DREAM as a QMS	
Activities	Tasks	Work Instructions	Forms
3. Verify SkN feasibility and consistency	3.1 Identify SkN relationships	W-4 Stakeholder Needs feasibility and consistency	F-4a Stakeholder Needs feasibility and consistency
	3.2 Stakeholder negociation		F-4b Stakeholder Negotiation

Once the stakeholder needs are elicited, captured, and consolidated, it is necessary to verify if the set of stakeholder needs are feasible and consistent, without contradictions among them, and compatibles.

Task 3.1: Identify Stakeholder Needs feasibility and consistency

Our Proposal:

We propose to do it in early stages (when they still are *stakeholder needs*), as stated by several authors as Haramis (1992), Bylund, Wolf, & Mazur, (2012), Mazur (2012), and Schillo, Isabelle, & Shakiba (2017) to avoid rework lately, and to reduce or eliminate waste of resources by focusing only in what is feasible. Some Systems Engineer's authors as Faisandier (2012) highlight the verification activity later when talking about stakeholders requirements. In addition, our proposition takes into account the lean enablers of Oehmen (2012):

- *"Proactively resolve potential conflicts between stakeholder values and expectations, and seek consensus*
- *The program manager must personally understand, clarify and remove ambiguity, conflicts, and waste from key requirements and expectations at the program start."*

How to do it:

We propose to develop this task through a collaborative session. The first step is to determine if *every pair* of stakeholder needs are related, and if so, determine how strong this co-relation is; it is suggested to use the following code (Hauser & Clausing, 1988):

++	Strong positive: direct and strong relationship between the system requirements
+	Medium positive: direct and weak relationship between the system requirements
_	Medium negative: indirect and weak relationship between the system requirements
_ _	Strong negative: indirect and strong relationship between the system requirements
(*nothing*)	If there is no co-relationship between the pair of system requirements)

During the collaborative session, the participants make consensus of all the pair of needs co-relationships. This way it is assured that the system will be feasible and consistent. Once all relationships are defined, the feasibility and consistency of every pair must be assessed. This way it is assured that the system will be feasible and consistent.
When a pair of stakeholder needs is not feasible nor consistent, the participants must negotiate until making consensus.

Task 3.2: Stakeholder negotiation

Our Proposal:

In case that one pair of SkN have negative (medium or strong) co-relation causing that the system is not feasible nor consistent (leading to a contradictory design), we propose that stakeholders negotiate until solve the negative co-relation between SkN.

How to do it:

We propose to do this task through a collaborative session due to the big impact in the future system.

We suggest that the stakeholders high weighted[8] make the decisions. In addition, we recommend the application of the Win-Win negotiation model (Narendhar & Anuradha, 2017):

a. Enter the Win conditions (stakeholder needs with negative (strong or medium) co-relationship)
b. Enter the Issues that summarize the involved conflict among the Win conditions
c. Prepare the candidate Options addressing every issue. The stakeholders evaluate the options to converge on a mutually satisfactory option
d. This Option is formally expressed in the Agreement schema

Every time that a stakeholder need is changed, the new stakeholder need co-relations must be identified; this task may become very hard due to the stakeholder interactions and long negotiation time consumed.

[8] The weight of every stakeholder was already obtained through the Activity 1, Task 1.2: Weight the Stakeholders

Activity 4: Prioritize Stakeholder Needs

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
4. Prioritize SkN	**4.1 Prioritize the retained SkN**	**W-5 Prioritize Stakeholder Needs**	**F-5 Prioritize Stakeholder Needs**
	4.2 Combine the prioritized and retained SkN with Stakeholder weights		
	4.3 Determine the relative weight of the resulting combination		

This activity includes three tasks: first, we propose to prioritize the retained stakeholder needs; later, combine the *stakeholder needs priority* (numerical value) with the *total stakeholder weights* (numerical value) to obtain the *stakeholder need value* to identify where *value* is; and finally, to determine the relative weight of the resulting combination.

Task 4.1: Prioritize the retained Stakeholder Needs

Our Proposal:

To prioritize the retained stakeholder needs we suggest applying the Analytical Hierarchy Process (AHP) developed by Saaty (2007).

This proposition integrates the suggestions of: a) Faisandier (2012, 2014), SEBoK (2016), Ryan et al. (2015) and Ryan & Faulconbridge (2016) regarding to prioritize the stakeholder needs; b) Bylund, Wolf, & Mazur (2012), Schillo, Isabelle, & Shakiba (2017), and Mazur (2009) when applying the Analytical Hierarchy Process (AHP) developed by Saaty (2007); and c) the contribution of Oehmen (2012) regarding the application of the lean enabler:

- *"Clarify, derive and prioritize requirements early, often and proactively"*

How to do it:

The AHP (Saaty, 2007) indicates that it is necessary to compare, by pairs, all the elements (*stakeholder needs* in our case) to define how they are co-related according to their importance or priority[9]. Table 57 illustrates the values to judge every pair of stakeholder needs.

Table 57 Comparative judgment table (Danesh & Ryan, 2015)

	Intensity Scale	
Less important than	Extremely less important	1/9
		1/8
	Very strong less important	1/7
		1/6
	Strongly less important	1/5
		1/4
	Moderately less important	1/3
		1/2
	Equal Importance	1
More important than		2
	Moderately more important	3
		4
	Strongly more important	5
		6
	Very strong more important	7
		8
	Extremely more important	9

[9] According to Badreau & Boulanger (2014) the prioritization criteria can be: the value of the business, the cost of implementation, the risk of implementation, or the penalty in case of rejection.

For example, if there are three stakeholder needs: *a, b*, and *c* they are going to be compared by pairs in a matrix; the intensity of their importance is assessed according to Table 57. Figure 80 illustrates the example.

	a	b	c
a	1	3*	1/2
b	1/3	1	7
c	2	1/7**	1

Figure 80 Matrix A that compares the priority of elements a, b, and c

The interpretation of the table for the pairs (a,b) and (c,b) is:

3*	Element *a* is moderately more important than *b*
1/7**	Element *c* is very strong less important than *b*

Once the pair comparison matrix is totally fulfilled, it is necessary to obtain the *total* of every column (by addition):

	a	b	c
a	1	3	1/2
b	1/3	1	7
c	2	1/7	1
total	3,33	4,14	8,5

Figure 81 Matrix A showing the total per column

To verify that all comparisons are consistent, the *consistency* of the matrix should be calculated. If this value is bigger than 0.10 it means that there are inconsistencies among the judgments. In that case, the value assignments (Matrix A) should be repeated until the *consistency* value is minor or equal to 0.10

Matrix B is obtained from the division of each element of Matrix A by the total value of its column:

Example: first column:

1 / 3.33 = 0.30

(1/3) / 3.33 = 0.10

2 / 3.33 = 0.60

Finally, the priority vector is constructed from the average of each Matrix B row (see Figure 82).

Example: first row: (0.30 + 0.72 + 0.06) / 3 = 0.36

	a	b	c	Priority vector
a	0,30	0,72	0,06	0,36
b	0,10	0,24	0,82	0,39
c	0,60	0,03	0,12	0,25
				1,00

Figure 82 Matrix B: normalized matrix

The interpretation of this results is: stakeholder need *b* is the most important (0,39 = 39%); stakeholder need *a* is the second on importance (0,36 = 36%); and finally, stakeholder need *c* is the least important (0,25 = 25%). The analysis and design team should take these results into account when designing the new system, to assign limited resources to priorities.

Once the Matrix A is completed, it is possible to use the online tool ANP SOLVER (free and available for academic purposes) at http://kkir.simor.ntua.gr/anpsolver.html (Rokou, 2009) to calculate the priority vector automatically. The provided package contains the software and user instructions.

Task 4.2: Combine the prioritized and retained stakeholder needs with stakeholder weights

Our Proposal:

We propose to combine stakeholder needs priority with the total stakeholder weights to discover where *value* is. This proposition is based on the knowledge that not all the stakeholders have the same power, legitimacy, and urgency; in this sense, one stakeholder need may be more important than another based on *who* expressed it.

How to do it:

In Activity 1 of the proposed methodology, we obtained the stakeholder weights; now the stakeholder needs have been mathematically prioritized. We propose to combine these two values to discover where the *value* is.

We propose to make this combination through multiplying:

Stakeholder need priority value x *Total stakeholder weight* = *Stakeholder need value*

NOTE: If one stakeholder need is expressed by more than one stakeholder, it should be taken into consideration the highest weight of the involved stakeholders at the time of making the combination.

The *stakeholder need values* can be ordered from maximum to minimum to better visualize the priorities.

Task 4.3: Determine the relative stakeholder need value of the resulting combination

Our Proposal:

The third step consists in determining the relative weight among the different *stakeholder need values (*Berx, Friedl, Witters & Hehenberger, 2016). This relative weight will help the analysis and design team and the project manager to visualize the advancement of the project in terms of the percentage of stakeholder requirements fulfillment.

How to do it:

It is necessary to add all *stakeholder need values*, and this *total* is 100%. Each *stakeholder need value* should be divided by the *total*.

Example: if considering *stakeholder need value* = 12, and *total* = 153, then the

Relative stakeholder need value = 12/153 = 0.07843

That corresponds to 7.843% of the stakeholder needs. If the stakeholder need is fulfilled, we can say that the system complies with 7.843% of total stakeholder needs.

NOTE: the *Relative Stakeholder Need Value* is a very useful and interesting data; in later stages, when the system prototype is built and ready to be validated, this data will help us to *quantify* if the system meets, or not, the stakeholder needs, making the validation process totally objective. We add value to the process (through systematic and objective prioritization) while reducing waste of time (generation of objective prototype validation parameters).

Activity 5: Transform Stakeholders Needs into Stakeholders Requirements

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
5. Transform SkN into Stakeholder Requirements (SkR)	**5.1 Transform the retained and prioritized SkN into SkR**	**W-6 List of Stakeholder Requirements**	**F-6 List of Stakeholder Requirements**
	5.2 SkR elaboration through decomposition and derivation		
	5.3 Assign a formal identifier to SkR		

At least, one stakeholder requirement must be generated from each stakeholder need. At present, there is not an automated tool capable of doing this translation automatically, human analysis is needed (Ryan, 2015).

The stakeholder requirements should be expressed in terms of the expected service and quality characteristics, and the language should be appropriate for the stakeholders to be understood.

Task 5.1: Transform the retained and prioritized stakeholder needs into stakeholder requirements

Our Proposal:

We suggest following the proven practices and recommendations proposed by several authors (for example Faisandier, 2012; Cuiller, 2015; Guide for writing requirements, 2017; Sukla, 2014) regarding how to write correctly stakeholder requirements. In addition, it could be possible to apply the method stated by Szejka et al. (2014), if software and hardware are available; finally, do not forget the recommendations of Oehmen (2012) regarding the application of lean enablers:

- *"Develop a robust process to capture, develop, and disseminate customer stakeholder value with extreme clarity*
- *Pursue a program vision and architecture that captures customer stakeholder requirements clearly and can be adapted to changes*
- *Use only highly experienced people and expert institutions to write program requirements, RPFs, and contracts*
- *If the customer lacks the expertise to develop clear requirements, issue a contract to a proxy organization with towering experience and expertise to sort out and mature the requirements and specifications in the RFP. This proxy must remain accountable for the quality of the requirements, including personal accountability*
- *Minimize the total number of requirements. Include only those that are needed to create value to the customer stakeholders*
- *Communicate to suppliers with crystal clarity all expectations, including the context and need, and all procedures and expectations for acceptance tests, and ensure the requirements are stable."*

How to do it:

We propose a collaborative session to analyze every stakeholder need. The analysis consists of the reflection of every stakeholder need statement to translate it into a stakeholder requirement statement.

The pattern for writing requirements is the following:

System/system component (subject) + Necessity (modal verb) +
Function/behavior/characteristics (verb phrase) + Condition (adjunct)

Examples:

The system shall provide first aid for injured people
The system shall be built using an existing industrial rolling base
The system shall be available 70% of the time

Recommendations:

- ✗ Avoid vague and general terms
- ✗ Avoid unbounded or ambiguous terms like:
  - ✗ Superlatives such as best or most
  - ✗ Subjective language such as user-friendly, easy to use, cost-effective
  - ✗ Vague pronouns such as it, this, that
  - ✗ Ambiguous adverbs and adjectives such as almost, always, significant, minimal
  - ✗ Open-ended, non-verifiable terms such as: provide support, or but not limited to.
  - ✗ Comparative phrases such as: better than, higher quality
  - ✗ Loopholes such as: if possible, as appropriate, as applicable
  - ✗ Incomplete references like not specifying the reference with its date and version number; or not specifying only the
  - ✗ Negative statements

Proven practices:

- ✓ Involve stakeholders early in the analysis
- ✓ Presence of rationale for each stakeholder requirement
- ✓ Analyze sources of stakeholder requirements before starting the definition of the system requirements
- ✓ Use of modeling techniques
- ✓ Use of requirements managing tool to trace linkages and to record the source of each stakeholder requirement
- ✓ Model the system of interest (SoI) from a higher-level system to identify and define its context (services, functional and physical interfaces, etc.)
- ✓ Use language and synonym dictionaries; semantics is the key to the correct expression of requirements
- ✓ When talking about a complex requirement, write complementary requirements that restrict the possible deviated interpretation of the original requirement.

The requirement statement should fulfill the next individual requirement quality characteristics: mature, accurate, appropriate to the level, complete, conforming, verifiable, necessary, singular, correct, unambiguous, feasible, implementation-free, and consistent; the stakeholder requirement should be traceable to the specific stakeholder need it came from.

With our proposal, we add value to the process and reduce waste of time since the stakeholder requirements are already classified by type; this is because the elicitation of stakeholder needs was done through the aid of the requirement classification.

Task 5.2: Stakeholder Requirement elaboration through decomposition and derivation

This task should be done with care, it requires human analysis and experience (Oehmen, 2012).

Our Proposal:

We suggest to combine the recommendations done by several authors; for example, in order to detect if there are implicit requirements, not expressed by the stakeholders but needed to build the system, the analysis and design team should infer what is missing; then, the *gemba* suggested by Mazur (2012) and Bylund, Wolf, & Mazur (2012) could be applied *in situ*; without these non-spoken requirements the system could not be able to exist. In addition, we suggest to follow lean enablers (Oehmen, 2012):

- *"Use only highly experienced people and expert institutions to write program requirements, and contracts*
- *If the customer lacks the expertise to develop clear requirements, issue a contract to a proxy organization with towering experience and expertise to sort out and mature the requirements and specifications. This proxy must remain accountable for the quality of the requirements, including personal accountability*
- *Always clearly link requirements to specific customer stakeholders needs and trace requirements from this top level to bottom level.*
- *Listen for and capture unspoken customer requirements"*

How to do it:

We propose the following questions to analyze each stakeholder requirement:

- Is the stakeholder requirement very complex?
- If the stakeholder requirement is split, it would have a better and correct treatment?
    - If the answers are "*yes*", then it is suitable to create decomposed stakeholder requirements
- Is there any related and necessary requirement that helps the feasibility of the system?
    - If the answers are "*yes*", then it is suitable to create derived stakeholder requirements

Task 5.3: Assign a formal identifier to stakeholder requirements

Our Proposal:

Systems Engineering authors as Faisandier (2014), Cuiller (2015), Ryan et al. (2015) and Ryan & Faulconbridge (2016) suggest assigning a formal identifier to stakeholder requirements for identification and future management.

We propose to assign to every stakeholder requirement a formal identifier as follows:

SkR – X.Y.Z – a - b

Where:

SkR	Stakeholder requirement
X	Number related to system requirement type classification group
Y	Number related to system requirement type classification sub-group (if available)
Z	Number related to system requirement type classification sub-sub-group (if available)
a	Consecutive number of stakeholder requirement
b	Consecutive number of derived or decomposed stakeholder requirement (if necessary)

For example, SkR – 4.2.0 - 1 – 1 is the formal identifier of the stakeholder requirement (SkR) of environmental conditions (4.2.0), is the first requirement of this type (1), and is a decomposed requirement (1).

NOTE: Following our proposal, traceability is being constructed automatically. For example, the stakeholder need *SkN – 4.2.0 - 1* is directly linked to the stakeholder requirement *SkR – 4.2.0 – 1*. We add value to the process (through providing identifiers systematically for the SkN and SkR) while reducing waste of time (trying to identify the parent's need for a requirement).

Activity 6: Draft Stakeholders Requirement document

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
6. Draft the SkR document	**6.1 Assign attributes to SkR**	**W-7 Stakeholder Requirements**	**F-7a Stakeholder Requirements**
	6.2 Identify potential risks		**F-7b Set of Stakeholder Requirements**
	6.3 Identify the stakeholders related to each SkR		

As mentioned before, this document will be useful later for managing purposes. Authors like Faisandier (2014), Cuiller (2015), and Ryan & Faulconbridge (2016) agree in the need of having automatized tools to draft requirements. The lean enablers suggested by Oehmen (2012) that should be followed are:

- *"Establish a plan that delineates the artifacts and interactions that provide the best means for drawing out stakeholder requirements*
- *Follow up written requirements with verbal clarification of context and expectations to ensure mutual understanding and agreement. Keep the record in writing, share the discussed items, and do not allow requirements creep"*

Task 6.1: Assign attributes to stakeholder requirements

Our Proposal:

We recommend to consider the requirement attributes suggested by Ryan et al. (2015) and Wheatcraft & Ryan (2018); the attributes are classified in four groups, nevertheless, at this level of complexity (SkN) it is suggested to take into account only three of them: 1) attributes to help define the requirement and its intent, 2) attributes associated with the system of interest (SoI) and system verification, and 3) attributes to help maintain the requirements. The attributes to show applicability and allow reuse will be included when talking about system Requirements (SyR).

How to do it:

According to literature, this task should be developed by the related stakeholders. We propose to work in a collaborative session to agree to the SkR attributes.

Task 6.2: Identify potential risks that could be generated by stakeholder requirements

Our Proposal:

As several authors recommend (Faisandier, 2014; Bylund, Wolf, & Mazur, 2009; SEBoK, 2016) it is imperative to conduct a rigorous identification of potential risk to avoid future problems in the process. Oehmen (2012) suggests to:

- *"Actively promote the maturation of Stakeholder Requirements, e.g., by providing detailed trade-off studies, feasibility studies and virtual prototypes"*

How to do it:

This technical risk analysis is based on some practices like:

a. Analysis of potential threats or undesired events and their probability of occurrence
b. Analysis of the consequences of these threats or undesired events classified by gravity
c. Mitigation to reduce the probabilities of threats and the levels of harmful effects to acceptable values (SEBoK, 2016)

We suggest following the proposition of Kossiakoff et al. (2011), Costello & Ferguson (2010), and Kehl et al. (2014): to document the risk and to develop the risk mitigation waterfall chart.

Task 6.3: Identify the stakeholders of each requirement

Our Proposal:

We propose to include the task of identifying the stakeholders of each requirement anticipating the requirement management process.

Requirement management process takes time since requirements are constantly evolving. Consequently, stakeholders must manage these changes through time. This management consumes one of the most precious resources of the project: time. Our proposal consists of the correct identification of the involved stakeholders for *each* requirement. This way; only the stakeholders implicated in the specific requirement will manage it.

The result: we add value to the process (through reducing the quantity of involved stakeholder in the management of a single requirement) and reduce waste of personal stakeholder time and effort. For sure, stakeholders will appreciate this part of DREAM methodology.

As suggested by Oehmen (2012), we consider the application of the following lean enablers:

- *"Frequently engage the stakeholders throughout the program life cycle*
- *Establish frequent and effective interaction with internal and external stakeholders*
- *Structure communication among stakeholders (who, how often, and what)*
- *When defining requirements sets for multiple suppliers, ensure that they are independent of each other, in order to minimize risk and reduce the need to manage dependencies among suppliers*
- *For non-routine tasks, avoid rework by coordinating task requirements with internal customer"*

It is highly recommended to use an effective communication tool to help the stakeholder interaction.

How to do it:

We propose to:

1. **Identify the author of every SkR.** Note: the author of the SkR is the author of its parent SkN. (Authors are available through Activity 2: Elicit, capture, and consolidate Stakeholder Needs, Task 2.2: Elicit stakeholder needs with aid of requirement classification)
2. **Identify co-related SkR.** Note: the co-related SkR can be identified by identifying their co-related SkN of the parent SkN. (Co-related SkN are available through Activity 3: Verify Stakeholders Needs feasibility and consistency, Task 3.1: Identify Stakeholder Needs feasibility and consistency)

3. **Identify the authors of the co-related SkR.** Note: the author of the co-related SkR is the author of the co-related SkN of the parent SkN. (Authors are available through Activity 2: Elicit, capture, and consolidate Stakeholder Needs, Task 2.2: Elicit stakeholder needs with aid of requirement classification)
4. **In addition, take into account the phase of the system life cycle where the stakeholders are involved whit this SkR.**

Activity 7: Verify and Validate Stakeholder Requirements

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
7. Verify and validate SkR	**7.1 Verify the quality of SkR**	**W-7 Stakeholder Requirements**	**F-7a Stakeholder Requirements**
	7.2 Validate SkR		**F-7b Set of Stakeholder Requirements**

Task 7.1: Verify the quality of Stakeholder Requirements

This task will help to assure that the stakeholder requirements will lead –in the near future- to the correct definition of system requirements.

Our Proposal:

As suggested by several authors (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015; Mokammel et al., 2018), we recommend to assess the statement of stakeholder requirements against the quality characteristics presented in *good* requirements, individually and as a set of requirements presented in Chapter 2.

As suggested by Haramis (1992), Mazur (2012), Bylund, Wolf, & Mazur (2009), and Schillo, Isabelle, & Shakiba (2017), in Activity 3: Verify Stakeholders Needs feasibility and consistency was already verified that there is no contradiction among the stakeholder needs, and the possible conflicts have already been solved.

The lean enablers suggested by Oehmen (2012) proposed to develop this activity are:

- *"Require an independent mandatory review of the program requirements, the concept of operation, and other relevant specifications of value for clarity, lack of ambiguity, lack of conflicts, stability, completeness, and general readiness for contracting and effective program execution*
- *Prevent careless insertion of mutually competing and conflicting requirements, excessive number of requirements, standards, and rules to be followed in the program, mindless "cut-and-paste" of requirements from previous programs*
- *Use a clear decision gate that reviews the maturity of requirements, the trade-offs between top-level objectives, and the level of remaining requirements risks before detailed formal requirements or a request for proposal is issued*
- *Match technologies to program requirements. Do not exceed program needs by using unnecessarily exquisite technologies ("gold plating")*
- *Perform robust system architecting and requirement analysis to determine technology needs and current technology readiness levels*
- *The program manager must personally understand, clarify and remove ambiguity, conflicts, and waste from key requirements and expectations al the program start.*
- *Align program metrics with intended benefits and stakeholder expectations."*

How to do it:

1. Once the requirement statement is built, it is necessary to verify that the requirement statement has maturity, is exhaustive, has accuracy, and is feasible; this verification can be done through the following questions (Faisandier, 2012):
   - **Maturity:** Is the expression of the stakeholder requirements close to the stakeholder expectations?
   - **Exhaustive:** Were all the stakeholders identified and interviewed? Were all the stakeholder requirements expressed and written?
   - **Accuracy:** Did the stakeholder express their expectations with precision?
   - **Feasible:** Was stakeholder requirement feasibility assessed through identified operational concepts?
2. Also, it is suggested to verify that it has the quality characteristics of a *good* requirement statement, as an individual requirement (Table 58) and as a set of requirements (Table 59); we propose the following questions:

Table 58 Questions to verify individual requirement quality characteristics

Individual Requirement Characteristics	Questions
Appropriate to level	Is the requirement appropriate to the level at which it is stated?
Complete	Does the requirement need further explanation? Is the SkR explained enough?
Conforming	Is the requirement conform to a standard formal structure?
Verifiable	Can be verified that the system meets or possesses the requirement?
Necessary	Should the system be able to function in the desired way with this requirement?
Singular	Is this requirement a combination of two or more requirements?
Correct	Is this the correct requirement that will result in the desired system performance?
Unambiguous	Is there only one interpretation of the requirement?
Feasible	Is the requirement achievable using existing technologies and manufacturing?
Implementation free	Does the requirement state what is required?
Consistent	Is the requirement free of conflicts with other requirements?

Table 59 Questions to verify quality characteristics of a set of requirements

Characteristics of a Set of Requirements	Questions
Complete	Does the set of requirements need no further amplification?
Consistent	Do the set of requirements have NOT individual requirements contradictory?
Affordable / Feasible	Can the set of requirements be satisfied by a solution that is feasible within life-cycle constraints?
Bounded	Do the set of requirements maintain the identified scope for the intended solution without increasing beyond what is needed?
Comprehensible	Is the set of requirements written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part?
Able to be validated	Is the set of requirements able to be proven the requirement set will lead to the achievement of the entity needs within the constraints?
Non-redundant	Do the set of requirements contain repeated individual requirements?

Task 7.2: Validate Stakeholder Requirements

Validation is a necessary and crucial activity, because it will determine if the analysis and design team: a) has found out the real and valuable stakeholder needs, and b) has been capable of translating them into stakeholder requirements. "*The voice of the customer should be heard*" (Blanchard & Fabrycky, 2006).

Our Proposal:

We suggest applying the recommendation of Faisandier (2012) to validate the stakeholder requirements through the *understanding, relevance,* and *justification.* At this time, we recommend, as stated by Mazur (2012), Bylund, Wolf, & Mazur (2009), to develop a competence study to identify if there is a system in the market that satisfy the need, or not, and in what degree. It is recommended to follow the lean enablers described by Oehmen (2012):

- *"Develop high-quality program requirements among customer stakeholders before the bidding and execution process begins*
- *Ensure that the customer-level requirements defined in the request for proposal (RFP) or contracts are truly representative of the need, stable, complete, crystal clear, de-conflicted, free of wasteful specifications, and as simple as possible*
- *Require personal and institutional accountability of the reviewers of requirements until program success is demonstrated*
- *Use peer-review requirements among stakeholders to ensure consensus validity and absence of conflicts*
- *Follow up written requirements with verbal clarification of context and expectations to ensure mutual understanding and agreement. Keep the record in writing, share the discussed items, and do not allow requirements creep*
- *Use architectural methods and modeling to create a standard program system representation (3D integrated CAE toolset, mockups, prototypes, models, simulations, and software design tools) that allow interactions with customers and other stakeholders as the best means of drawing out requirements*
- *Fail early and fail often through rapid learning techniques (e.g., prototyping, test, simulations, digital models or spiral development)*
- *Communicate to suppliers with crystal clarity all expectations, including the context and need, and all procedures and expectations for acceptance tests, and ensure the requirements are stable*
- *Be willing to challenge the customer's assumptions on technical and meritocratic grounds, and to maximize program stability, relying on technical expertise*
- *Align program metrics with intended benefits and stakeholder expectations."*

How to do it:

Once every stakeholder requirement has been verified, it is time to validate them. The stakeholder(s) responsible for validation should be defined; for example the owner or investor, the project executive or project sponsor, consumers, or users.

The validation of every stakeholder requirement statement can be done through the following criteria:

- **Understanding:** each stakeholder owning the requirement must validate that it is understandable, and confirm that the resolution of the conflict with other requirements does not compromise their intentions

The validation of the set of stakeholder requirements is suggested to do it through the following criteria:

- **Relevance:** the expression of the requirements allows to define the importance of the solution?

- **Justification:** Why do these needs and expectations exist? What risk or cause could make these expectations disappear?

Activity 8: Record and manage Stakeholder Requirements

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
8. Record and manage SkR	**8.1 Record SkR**	**W-7 Stakeholder Requirements**	**F-7a Stakeholder Requirements**
	8.2 Manage SkR evolution or modification		**F-7b Set of Stakeholder Requirements**

Task 8.1: Record stakeholder requirements

Our Proposal:

We suggest recording the stakeholder requirements with the aid of a support tool.

How to do it:

We recommend to use a support tool to record every stakeholder requirement, its attributes, identifier, its parent need and co-related stakeholder requirements (to maintain backward and forward traceability).

Every filled form will become a *record*, part of the quality management system that will assure that the stakeholder requirements were defined.

Task 8.2: Manage stakeholder requirements evolution or modification

Our Proposal:

We suggest managing the stakeholder requirements with the aid of a support tool.

How to do it:

Needs and requirements are continuously evolving. Every time that any stakeholder need or requirement is added, modified, or canceled, this action should be recorded, to assure that everything is being documented and visible for the analysis and design team, and the rest of stakeholders.

At this point, it is necessary to remember that the stakeholder needs and stakeholder requirements are inter-related. DREAM methodology allows iterations when required, it is, when a stakeholder need or requirement is changed, modified, or canceled, the DREAM activities and tasks should be repeated as necessary to assure that all changes and its consequences have been assessed, verified and validated. Table 60 illustrates what activities and tasks should be performed in case of a stakeholder need or requirements is added, modified, or canceled.

Table 60 Activities and task needed to manage stakeholder requirements

DREAM methodology		If the stakeholder need or requirement is :		
Activities	**Tasks**	**Added**	**Modified**	**Canceled**
1. Identify and Weight Stakeholders	**Identify the Stakeholders across the life cycle**			
	Weight the Stakeholders			

2. Elicit, capture, and consolidate Stakeholder Needs (SkN)	**Elicit, capture, and consolidate the SkN, expectations, objectives, and constraints**	✓		
	Elicit SkN with the aid of requirement classification			
3. Verify SkN feasibility and consistency	**Identify SkN relationships**	✓	✓	✓
	Stakeholder negotiation	✓	✓	✓
4. Prioritize SkN	**Prioritize the retained SkN**	✓		✓
	Combine the prioritized and retained SkN with Stakeholder's weights	✓		
	Determine the relative weight of the resulting combination	✓		✓
5. Transform SkN into Stakeholder Requirements (SkR)	**Transform the retained and prioritized SkN into SkR**	✓	✓	
	SkR elaboration through decomposition and derivation	✓	✓	
	Assign a formal identifier to SkR	✓		
6. Draft the SkR document	**Assign attributes to SkR**	✓		
	Identify potential risks	✓	✓	✓
	Identify the Stakeholders related to each SkR	✓		
7. Verify and validate SkR	**Verify the quality of SkR**	✓	✓	
	Validate SkR	✓	✓	
8. Record and manage SkR	**Record SkR**	✓	✓	✓
	Manage SkR evolution or modification	✓	✓	✓

6.3 From Stakeholder Requirements to System Requirements process

Figure 83 shows the flowchart of the proposed process to transform *stakeholder requirements into system requirements*; on the left side are shown the process activities, and on the right side the tasks. This process is iterative; it means that at any time it could be done again and again in order to improve the results.

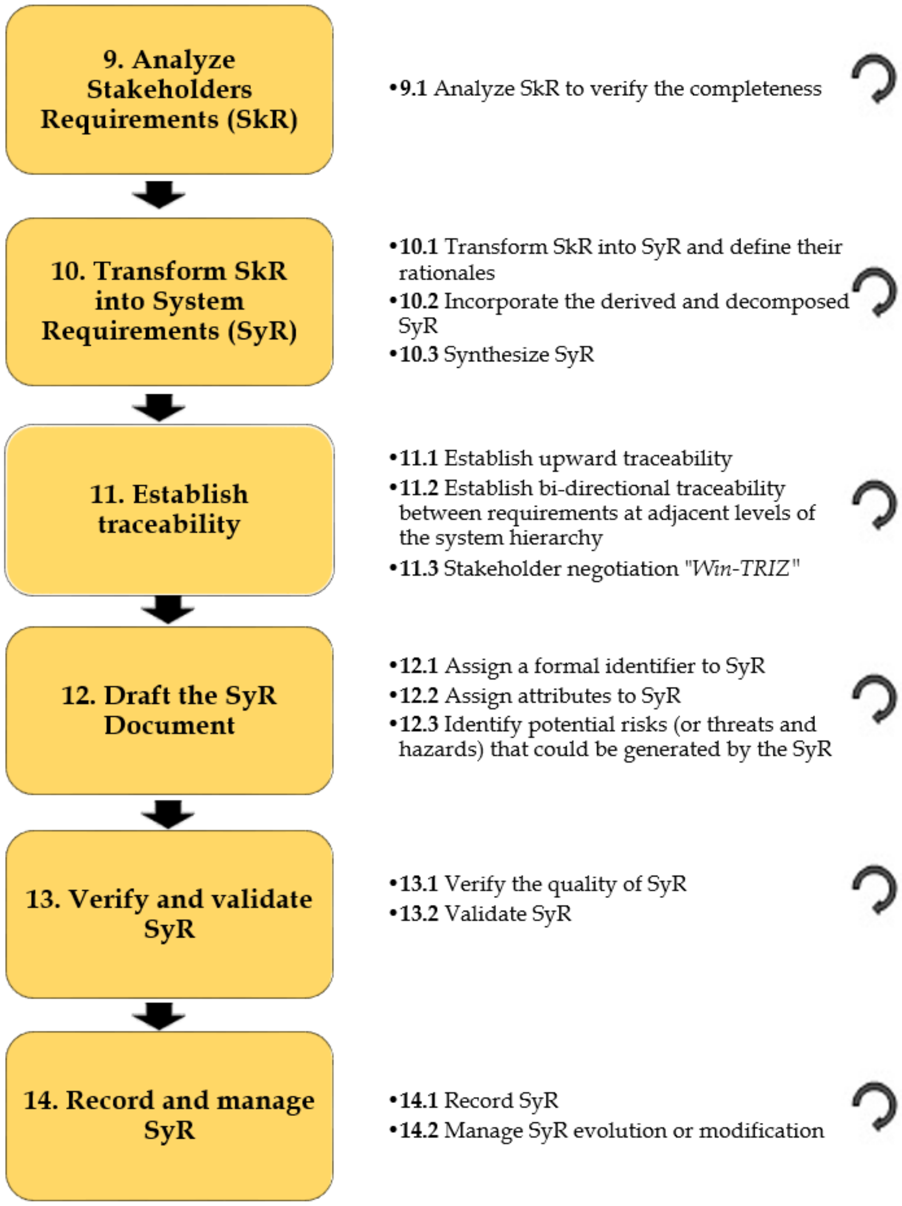


Figure 83 Proposal: stakeholder requirements to system requirements process

Table 61 shows the relationship among the activities and task of the proposed process to transform *stakeholder needs into stakeholder requirements*, related to the work instructions and forms of DREAM as a QMS. See Annex 3.

Table 61 Relationship among methodology and DREAM as a QMS

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
9. Analyse Stakeholders Requirements (SkR)	**9.1 Analyse SkR to verify completeness**	**W-8 Analyse Stakeholder Requirements**	**F-8 Analyse Stakeholder Requirements**
10. Transform SkR into System Requirements (SyR)	**10.1 Transform SkR into SyR and define the rationals**	**W-9 List of System Requirements**	**F-9 List of System Requirements**
	10.2 Incorpore the derived and decomposed SyR		
	10.3 Synthesise SyR		
11.Establish traceability	**11.1 Establish the upward traceability with the SkN and SkR**	**W-10 Traceability at adjacent levels of System Requirements**	**F – 10a Traceability at adjacent levels of System Requirements**
	11.2 Establish bi-directional traceability between SyR at adjacent levels of the system hierarchy		**F-10b Stakeholder Negotiation**
	11.3 Stakeholder negotiation"Win-TRIZ"		
12. Draft the SyR Document	**12.1 Assign a formal identifier to SyR**	**W-11 System Requirements**	**F-11a System Requirements**
	12.2 Assign attributes to SyR		
	12.3 Identify potential risks (of threats and hazards) that could be generated by the SyR		
13. Verify and validate SyR	**13.1 Verify the quality of SyR**		**F-11b Set of System Requirements V&V**
	13.2 Validate SyR		
14. Record and manage SyR	**14.1 Record SyR**		
	14.2 Manage SyR evolution or modification		

Following subsections explain, for each activity and its tasks:

- *Our proposal*: this section shows the bibliographical support and proposition
- *How to do it*: this section explains how to develop the task.
- The *support documents:* work instructions (W) and forms (F) that are part of DREAM as a QMS.

Activity 9: Analyze Stakeholders Requirements (SkR)

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
9. Analyse Stakeholders Requirements (SkR)	**9.1 Analyse SkR to verify completeness**	**W-8 Analyse Stakeholder Requirements**	**F-8 Analyse Stakeholder Requirements**

Tasks 9.1: Analyze stakeholder requirements to verify the completeness

Our Proposal:

Following the recommendation of Faisandier (2012), we suggest analyzing if:

- The list of stakeholders is complete and they are already involved in the project
- As far as possible, all the possible scenarios are considered
- As far as possible, other life-cycle stages scenarios are already defined
- The stakeholder requirements are consolidated and organized

How to do it:

We propose to develop this activity through a collaborative session. To perform these tasks it is necessary to analyze the following information contained in records:

1. Verify that, as far as possible, the involved stakeholders are being identified and involved. In case that one or more stakeholders are identified, they should be added to records, and involved in the project. We propose the next question:
    - Are the stakeholders through the system life cycle being identified?
2. Verify that, as far as possible, the possible scenarios have been considered, as well the assumptions, what is the "value" for the stakeholders, and what is the "success" for the project, the system context of use, the functional and physical interfaces of the system, the input-output flows of material, energy and/or information, the necessary physical connections to carry the exchanged flows of material, energy and/or information. In case that some stakeholders are added into the project, they should be able to review and contribute with their ideas to complete the records, and the person responsible for validation should agree. We propose the next questions:
    - Have the different scenarios been identified?
    - Are the assumptions being expressed and understood?
    - Has been identified where the *value* is?
    - Has been identified what *success* is for the project?
    - Have been identified the functional and physical interfaces of the system?
    - Have been identified the flows of material, energy and/or information?
3. In case that some stakeholders are added into the project, they should be able to review and contribute with their ideas to complete the records, and the person responsible for validation should agree.
4. Verify, as far as possible, the stakeholder needs are identified and translated into stakeholder requirements, as well as the constraints; for example the enterprise context, procedures, environmental conditions, programmatic constraints, available resources and available technologies, interfaces with existing systems, etc.

Activity 10: Transform SkR into System Requirements (SyR)

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
10. Transform SkR into System Requirements (SyR)	**10.1 Transform SkR into SyR and define the rationals**	**W-9 List of System Requirements**	**F-9 List of System Requirements**
	10.2 Incorpore the derived and decomposed SyR		
	10.3 Synthesise SyR		

Task 10.1: Transform SkR into SyR and define their rationales

To transform SkR into SyR is a *human work*, where reflection and thinking are combined with communication skills. To perform this task it is necessary to translate *natural language* in a more *technical* one. Experience is needed, and if the design and analysis team lacks this expertise, authors recommend to ask for help.

Our Proposal:

We recommend following proven practices when writing system requirements.

How to do it:

We propose to develop this activity through a collaborative session.

- ✓ Check that stakeholder requirements are complete before the definition of the system requirements (Faisandier, 2012; Badreau & Boulanger, 2014; Ryan et al., 2015; Ryan & Faulconbridge, 2016; SEBoK, 2016)
- ✓ Pay particular attention to the system's "inputs" and "outputs" (Cuiller, 2015)
- ✓ Focus on *key* requirements (Hull, Jackson, & Dick, 2011; Cuiller, 2015)
- ✓ Involve the stakeholders as soon as possible while developing system requirements (Faisandier, 2012; Oehmen, 2012; SEBoK, 2016)
- ✓ Use short sentences (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Use of modeling techniques (Hull, Jackson, & Dick, 2011; Faisandier, 2012; SEBoK, 2016; Ryan & Faulconbridge, 2016)
- ✓ Paraphrasing stakeholder requirements (Cuiller, 2015)
- ✓ Rethinking stakeholder requirements in a technical language (Cuiller, 2015)
- ✓ One concept should be determined by one or several terms and should be used everywhere as this term or this set of terms (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Define a glossary for words having a general meaning different of the technical meaning (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Translate a single idea into a single requirement (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Begin the sentences with: "*The system should*... or *The system shall*...." (Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015)
- ✓ Requirements should be written in simple sentences: subject + verb in present tense + complement (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Review that requirements meet the quality characteristics of "system requirements" and "set of system requirements". (Hull, Jackson, & Dick, 2011; Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015; Mokammel et al., 2018)
- ✓ Justifying the requirements through a rationale. (Faisandier, 2012; Badreau & Boulanger, 2014; Cuiller, 2015; Ryan, 2015; Ryan & Faulconbridge, 2016; SEBoK, 2016)
- ✓ Requirements should be SMART (Specific, Measurable, Achievable, Realistic, Testable) (Faisandier, 2012),
- ✓ Organize peer reviews with applicable experts (Faisandier, 2012; SEBoK, 2016)
- ✓ Use typical measures for requirements engineering for volatility, trends, verification progress, validation progress, and peer review defects (Faisandier, 2012; SEBoK, 2016)
- ✓ Consider using a requirements management tool to trace linkages between system requirements and to display their relationships (Faisandier, 2012; SEBoK, 2016)
- ✓ Ensure that all types of requirements are considered (Faisandier, 2012; Cuiller, 2015)

Issues about wording:

- ✓ Adjectives should be measurables and quantifiables to allow verification (Faisandier, 2012)
- ✓ Avoid sentences with double negation (Faisandier, 2012; Badreau & Boulanger, 2014),
- ✓ Avoid orienting the solution (Faisandier, 2012)
- ✓ Avoid subjective sentences (Faisandier, 2012; Badreau & Boulanger, 2014)
- ✓ Avoid indefinite, generic or meaningless terms (Faisandier, 2012)
- ✓ Avoid speculation (Hull, Jackson, & Dick, 2011)
- ✓ Avoid wishful thinking (example: 100% reliable) (Hull, Jackson, & Dick, 2011)
- ✓ Avoid vague adjectives as: some, any, different, various, several, fast, correct, good, serious, and critical (Hull, Jackson, & Dick, 2011; Faisandier, 2012)
- ✓ Avoid vague adverbs as: little, much, enough, less, more, after, front, often, sometimes, a long time, at once, quickly, well, almost, and correctly (Hull, Jackson, & Dick, 2011; Badreau & Boulanger, 2014)
- ✓ Consult the dictionary as frequently as possible (Faisandier, 2012)
- ✓ Avoid common drafting errors such as contamination, redundancy, implicit requirements, inaccuracies, contradiction, ambiguities or omissions (Hull, Jackson, & Dick, 2011; Faisandier, 2012)

Task 10.2: Incorporate the derived and decomposed SyR

Our Proposal:

We recommend to follow the suggestions of SEBoK (2016), Faisandier (2012) and Ryan & Faulconbridge (2016): to use the type classification of assignments to help the designers and analysts to know where the requirement comes from (direct assignment, Indirect assignment -simply decomposed or modeled and decomposed-, or derived requirement from design).

How to do it:

We propose to develop this activity through a collaborative session.

1. Once the design and analysis team has the list of system requirements, it is necessary to identify if derived or decomposed system requirements are needed. The following questions may be useful to conduct this identification:
   - Does this requirement come from the stakeholder need elicitation process?
   - Does this requirement come from more complex requirements which are decomposed?
   - Does this requirement come from more complex requirement which is decomposed & modeled?
   - Does this requirement come from architecture or design decisions?
2. If necessary, add the derived or decomposed system requirements to the list of system requirements

Task 10.3: Synthesize SyR

Our Proposal:

We suggest identifying those requirements that are duplicated (SEBoK, 2016) to synthesize the list of system requirements.

How to do it:

We propose to develop this activity through a collaborative session.

1. Ask the question: *Are there duplicated system requirements?*
2. Read the records to verify
3. If yes, write this observation in the record to not consider anymore the duplicated system requirement

Activity 11: Establish traceability

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
11.Establish traceability	**11.1 Establish the upward traceability with the SkN and SkR**	**W-10 Traceability at adjacent levels of System Requirements**	**F – 10a Traceability at adjacent levels of System Requirements**
	11.2 Establish bi-directional traceability between SyR at adjacent levels of the system hierarchy		**F-10b Stakeholder Negotiation**
	11.3 Stakeholder negotiation"Win-TRIZ"		

Task 11.1: Establish the upward traceability with the stakeholders needs (SkN) and stakeholders requirements (SkR)

Our Proposal:

With our proposal traceability is done automatically. The identifiers proposed before help maintaining the traceability among stakeholder needs, stakeholder requirements, and system requirements.

How to do it:

Every SkN has a unique identifier, consequently, each SkR and SyR also have unique identifiers directly related to their parent need.

Identifier for stakeholder needs:	SkN – X.Y.Z – a
Identifier for stakeholder requirements:	SkR – X.Y.Z – a - b
Identifier for system requirements:	SyR – X.Y.Z – a - b

Where:

SkN	Stakeholder need
SkR	Stakeholder requirement
SyR	System requirement
X	Number related to requirement type classification group
Y	Number related to requirement type classification sub-group (if available)
Z	Number related to requirement type classification sub-sub-group (if available)
a	Consecutive number of need or requirement
b	Consecutive number of derived or decomposed requirement (if available)

NOTE: Following our proposal, traceability is being constructed automatically. For example, the stakeholder need *SkN – 4.2.0 - 1* is directly linked to the stakeholder requirement *SkR – 4.2.0 – 1* and the system requirement *SyR – 4.2.0 – 1*. We add value to the process (through providing identifiers systematically for the SkN, SkR, and SyR) while reducing waste of time (trying to identify the parent's need for a requirement).

Task 11.2: Establish bi-directional traceability between requirements at adjacent levels of the system hierarchy

No bibliographic reference, other than SEBoK (2016), was found to explain the term "*bi-directional traceability between requirements at adjacent levels of the system hierarchy*". That is the reason why we propose the following definition:

> **Bi-directional traceability between requirements at adjacent levels of the system hierarchy:** is the association among two or more logical entities that is discernible at adjacent levels of the system hierarchy.

Taking this definition into account, the establishment of *bi-directional traceability between requirements at adjacent levels of the system hierarchy* could be done through the identification of what requirements at adjacent levels are co-related.

The importance of detecting *bi-directional traceability between requirements at adjacent levels of the system hierarchy* lies in the premise that when a requirement is modified, canceled or evolved, it will directly affect those requirements to which it is co-related.

As stated in Chapter 4, QFD allows to know the co-relationships between the system requirements already defined (Mazur, 2012; Yamashina, 2002); that is, through this method, it can be known if the different requirements are co-related. Consequently, with the aid of this matrix, it could be possible to identify the *bi-directional traceability between requirements at adjacent levels of the system hierarchy.*

Our Proposal:

To perform this activity we suggest following the method of co-relationship matrix used in the framework of QFD (Mazur, 2012). If available, we suggest applying knowledge-based semi-automated aids, like QARCC (Quality Attribute Risk and Conflict Consultant) and S-COST (Software Cost Option Strategy Tool) that provide improvements in conflict identification and resolution (Hoh In, Boehm, Rodger, & Deutsch, 2001).

How to do it:

We propose to develop this activity through a collaborative session.

1. The first step is to determine if *every pair* of system requirements are co-related, and if so, determine how strong this co-relation is; we suggest to use the following code (Hauser & Clausing, 1988):

++	Strong positive: direct and strong relationship between the system requirements
+	Medium positive: direct and weak relationship between the system requirements
_	Medium negative: indirect and weak relationship between the system requirements
_ _	Strong negative: indirect and strong relationship between the system requirements
(*nothing*)	If there is no co-relationship between the pair of system requirements

2. During the collaborative session, the participants make consensus of all the pair of requirements co-relationships. This way it is assured that the system will be feasible and consistent.
3. When a pair of system requirements is identified not consistent or contradictory, we propose the Task: Stakeholder Negotiation "*Win-TRIZ*"

Task 11.3: Stakeholder negotiation "Win-TRIZ"

Our Proposal:

We propose to develop this task in a collaborative session due to the big impact of the future system. We suggest that the stakeholders *high weighted* make the decisions.

Taking into account several contributions as:

a) WinWin negotiation model (Hoh In, Boehm, Rodger, & Deutsch, 2001; Narendhar & Anuradha, 2017);
b) Combine QFD with TRIZ (Yamashina, Ito, & Kawada, 2002);
c) Considering propositions made by Malinin (2016) at the moment of restatement of the initial problem (considering the different locations of the system, the different life stages of the system, and the different conditions of the system at different times); and finally
d) the suggestion of Kraev (2006) of separate the opposing physical contradictions and remove the combined area of their interaction,

Table 62 shows a comparison between the steps of Win-Win model and the TRIZ method. Both of them start with the identification of the contradictory system requirements; follow with the definition of the problem; at this point, TRIZ proposes a deeper analysis through the separation of opposite physical contradictions (numerically and/or graphically), continuing with the replacement of these quality characteristics by physical characteristics helped by the Altshuller's Contradiction Matrix (Altshuller, 2000); later, both methods propose the candidate options that possibly solve the system conflict; finally, the Win-Win negotiation model guides the stakeholders to evaluate the options that converge on a mutually satisfactory option that will be expressed and recorded in the Agreement schema.

Table 62 Comparison between Win-Win negotiation model and the TRIZ method

Win-Win (Hoh In, Boehm, Rodger, & Deutsch, 2001; Narendhar & Anuradha, 2017)	Kraev (2006)
Enter the Win conditions: system requirements with negative (strong or medium) co-relationship	Select the combined pair of system requirements (quality characteristics) that show a negative co-relation
Enter the Issues that summarize the involved conflict among the Win conditions	Restate of the initial problem.
	Separate the opposite physical contradictions i. Numerically, according to Malinin (2016) considering: a) the different locations of the system, b) the different life stages of the system, and c) the different conditions of the system at different times. ii. Graphically, according to Kraev (2006) considering: a) separation in time, b) separation in space, c) system transformations, and d) phase transformations, or physical-chemical transformation of substances If the contradiction remains…
	Replace the quality characteristics with general physical characteristics
	Refers to Altshuller's Contradiction Matrix

Prepare the candidate Options addressing every issue.	Derive from among the 40 principles those principles that possibly resolve the system conflict
The stakeholders evaluate the options to converge on a mutually satisfactory option	
This Option is formally expressed in the Agreement schema	

As it can be seen, both of them (Win-Win and TRIZ) complement each other; so, we decided to create and we propose a new method that we named "Win-TRIZ" that combines the Win-Win negotiation model with TRIZ method; both methods combined result in a stronger approach.

How to do it:

1. Enter the Win conditions, it is, the combined pair of SyR (quality characteristics) that show a negative co-relation (strong or medium)
2. Enter the Issues that summarize the involved conflict among the Win conditions
3. Restate the initial problem
4. Separate the opposite physical contradictions:
    a. Numerically, according to Malinin (2016) considering: a) the different locations of the system, b) the different life stages of the system, and c) the different conditions of the system at different times.
    b. Graphically, according to Kraev (2006) considering: a) separation in time, b) separation in space, c) system transformations, and d) phase transformations, or physical-chemical transformation of substances

    If the contradiction remains…
5. Replace the quality characteristics with general physical characteristics
6. Refers to Altshuller's Contradiction Matrix (Altshuller, 2000)
7. Derive from among the 40 principles those principles that possibly resolve the system conflict
8. Prepare the candidate options addressing every issue.
9. The Stakeholders evaluate the options to converge on a mutually satisfactory option
10. The Option is formally expressed in the Agreement Schema

Every time that a system requirement is changed, the new system requirement co-relations must be identified; this task may become very hard due to the stakeholder interactions and long negotiation time consumed.

Activity 12: Draft the SyR Document

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
12. Draft the SyR Document	**12.1 Assign a formal identifier to SyR**	**W-11 System Requirements**	**F-11a System Requirements**
	12.2 Assign attributes to SyR		
	12.3 Identify potential risks (of threats and hazards) that could be generated by the SyR		
	14.2 Manage SyR evolution or modification		

Task 12.1: Assign a formal identifier to SyR

Our Proposal:

As it has been said before, we propose the identifiers for stakeholder needs, stakeholder requirements, and system requirements. The assignment of formal identifiers is proposed as following:

How to do it:

ID for stakeholder needs:	SkN – X.Y.Z – a
ID for stakeholder requirements:	SkR – X.Y.Z – a - b
ID for system requirements:	SyR – X.Y.Z – a - b

Where:

SkN	Stakeholder need
SkR	Stakeholder requirement
SyR	System requirement
X	Number related to requirement type classification group
Y	Number related to requirement type classification sub-group (if available)
Z	Number related to requirement type classification sub-sub-group (if available)
a	Consecutive number of need or requirement
b	Consecutive number of derived or decomposed requirement (if available)

Task 12.2: Assign attributes to SyR

Our Proposal:

We recommend to consider the requirement attributes suggested by Ryan et al. (2015) and Wheatcraft & Ryan (2018), they are considered here as the most complete set of attributes; they are classified in four groups: 1) attributes to help define the requirement and its intent, 2) attributes associated with the system of interest (SoI) and system verification, 3) attributes to help maintain the requirements, and 4) attributes to show applicability and allow reuse.

How to do it:

According to literature, this task should be developed by the stakeholders that are related in a specific stakeholder requirement. It is suggested that they work in a collaborative session to agree on the stakeholder requirement attributes.

Task 12.3: Identify potential risks (or threats and hazards) that could be generated by the SyR

Our Proposal:

As authors like Faisandier, 2014; Bylund, Wolf, & Mazur, 2009; SEBoK, 2016 suggest, it is imperative to conduct a rigorous identification of potential risk to avoid future problems in the process.

How to do it:

This technical risk analysis is based on some practices like:

d. analysis of potential threats or undesired events and their probability of occurrence
e. analysis of the consequences of these threats or undesired events classified by gravity
f. mitigation to reduce the probabilities of threats and the levels of harmful effects to acceptable values (SEBoK, 2016)

We suggest following the proposition of Kossiakoff et al. (2011), Costello & Ferguson (2010), and Kehl et al. (2014): to document the risk and to do the Risk Mitigation waterfall chart.

Activity 13: Verify and validate SyR

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
13. Verify and validate SyR	**13.1 Verify the quality of SyR**	**W-11 System Requirements**	**F-11b Set of System Requirements V&V**
	13.2 Validate SyR		

Task 13.1: Verify the quality of SyR

Our Proposal:

We propose to verify the traceability: every SkN should be translated correctly into a SkR, and the resulting SkR should be translated correctly into a SyR; to demonstrate that:

a) the initial SkN has been taken into account;
b) the individual SyR and the set of SyR meet the quality criteria; and
c) no traceable requirements have been introduced (Badreau & Boulanger, 2014).

We recommend following Faisandier (2012) and SEBoK (2016) suggestions: verifying the SyR statements against the quality characteristics of a *good* requirement statement, as an individual requirement and as a set of requirements.

How to do it:

1. Verify that every SyR comes from a specific SkR, and at the same time, this SkR comes from a specific SkN. It is, that each stakeholder requirement has been translated into at least one system requirement.

Verify that the SyR statements meet the quality characteristics of a good *requirement statement, as an individual requirement (Table 63) and as a set of requirements (*

Table 64); we propose the following questions:

Table 63 Questions to verify individual requirement quality characteristics

Individual Requirement Characteristics	Questions
Appropriate to level	Is the requirement appropriate to the level at which it is stated?
Complete	Does the requirement need further explanation?
Conforming	Is the requirement conform to a standard formal structure?
Verifiable	Can be verified that the system meets or possesses the requirement?
Necessary	Should the system be able to function in the desired way with this requirement?
Singular	Is this requirement a combination of two or more requirements?
Correct	Is this the correct requirement that will result in the desired system performance?
Unambiguous	Is there only one interpretation of the requirement?
Feasible	Is the requirement achievable using existing technologies and manufacturing?
Implementation free	Does the requirement state what is required?
Consistent	Is the requirement free of conflicts with other requirements?

Table 64 Questions to verify quality characteristics of a set of requirements

Characteristics of a Set of Requirements	Questions
Complete	Does the set of requirements need no further amplification?
Consistent	Do the set of requirements have NOT individual requirements contradictory?
Affordable / Feasible	Can the set of requirements be satisfied by a solution that is feasible within life-cycle constraints?
Bounded	Do the set of requirements maintain the identified scope for the intended solution without increasing beyond what is needed?
Comprehensible	Is the set of requirements written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part?
Able to be validated	Is the set of requirements able to be proven the requirement set will lead to the achievement of the entity needs within the constraints?
Non-redundant	Do the set of requirements contain repeated individual requirements?

Task 13.2: Validate SyR

Our Proposal:

According to Kossiakoff et al. (2011), the system requirement validation may be performed *formally* (with the participation of an external entity) or *informally* (with the participation of customers and/or users). In this context, our proposal is located within the domain of *informal* validation done by customers and/or users.

Our proposition includes the suggestion of Faisandier (2012) and SEBoK (2016) regarding the use of the validation traceability matrix that relates stakeholder requirements to system requirements.

In addition, we consider the point of view of Badreau & Boulanger (2014) related to validation activities such as concept document review, prototypes and/or models. It is taken into account the main activities suggested by SEBoK (2016) as the establishment of a validation strategy, carrying out the validation through models or prototypes, comparing the obtained results versus the expected ones, and finally, the control process to perform corrective actions based on the non-conformities detected, process suggested by Boehm (1979).

How to do it:

We propose the following process:

1. The stakeholders who play the role of customers and/or users can validate that every SyR should meet a SkN.
2. To validate every individual SyR: the first question to be answered is:
    a. Does *this* SyR reflect *its* parent SkR, and *its* parent SkN?
3. There are some attributes that should allow the validation of the individual SyR, for example validation method, acceptance criteria, estimated cost, priority (*value*), urgency, risk, and quality characteristics
4. To validate the set of SyR:
    a. If a prototype or model is built: Does the prototype or model meet the stakeholder needs?

b. Are the system mission, purpose, objectives, and stakeholder needs congruent with the set of SyR?
c. Analyze the comparison between the results *obtained* versus the *expected* ones
d. In case of detecting a non-conformity:
   i. Isolate the source of the problem and develop a solution resulting in an iteration of SyR
   ii. Approve any proposed iteration of SyR which would perceptibly change the SyR baseline
   iii. Analyze the iteration of SyR and, in case of detecting any problem,
   iv. Continue with the verification and validation activities until the problems are fixed.

Activity 14: Record and manage SyR

Methodology		DREAM as a QMS	
Activities	**Tasks**	**Work Instructions**	**Forms**
14. Record and manage SyR	**14.1 Record SyR** **14.2 Manage SyR evolution or modification**	**W-11 System Requirements**	**F-11b Set of System Requirements V&V**

Task 14.1: Record SyR

Our Proposal:

The proposition is the use of a support tool, and to have a record for every single SyR.

How to do it:

Through filling forms using a support tool.

Task 14.2: Manage SyR evolution or modification

Our Proposal:

We propose to manage the system requirements with the aid of a support tool, in order to record every change or evolution on system requirements.

How to do it:

Every time that any system requirement is added, modified, or canceled, this action should be recorded in the forms, to assure that everything is being documented and visible for the stakeholders, analysis and design team.

The methodology allows iterations when required, it is, when a SyR is changed, modified, or canceled, the activities should be repeated as necessary to assure that all changes and its consequences have been assessed, verified and validated. Table 65 illustrates what activities and tasks should be performed in case of a system requirement is added, modified, or canceled.

Table 65 Activities and task needed to manage system requirements

		If the stakeholder need is :		
Activities	**Tasks**	**Added**	**Modified**	**Canceled**
9. Analyze Stakeholders Requirements (SkR)	**Analyze SkR to verify the completeness**	✓	✓	✓
10. Transform SkR into System Requirements (SyR)	**Transform SkR into SyR, define the rationals**	✓	✓	
	Incorporate the derived and decomposed SyR	✓	✓	
	Synthesize SyR	✓	✓	
11.Establish traceability	**Establish the upward traceability with the SkN and SkR**	✓		
	Establish bi-directional traceability between SyR at adjacent levels of the system hierarchy	✓	✓	✓
	Stakeholder negotiation	✓	✓	✓
12. Draft the SyR Document	**Assign a formal identifier to SyR**	✓		
	Assign attributes to SyR	✓		
	Identify potential risks	✓	✓	
13. Verify and validate SyR	**Verify the quality of SyR**	✓	✓	✓
	Validate SyR	✓	✓	✓
14. Record and manage SyR	**Record SyR**	✓	✓	✓
	Manage SyR evolution or modification	✓	✓	✓

6.4 Discussion

The most common cause of failure of change initiatives is resistance to change: "*Resistance to change is directly proportional to the magnitude of change and the time available to implement it*" (Niebel & Freiwalds, 2009).

Involving employees, showing them the benefits of the change, giving them training and time to assimilate it, will result in its successful application (Niebel & Freiwalds, 2009).

Based on this fact, change from one methodology for new systems development to another one implies a big change inside any organization. The administration or management should correctly implement any methodology in their work teams if they want to succeed in the challenge.

The analysis and design team, as well as the stakeholders, need to be trained before applying DREAM methodology. Bylund, Wolf, & Mazur, (2009) mention the importance of an educational package that accompanies the company during the new system development process. That is, the process, methods, and tools that the company decides to apply within its organization should have sufficient support and material to train personnel.

Such training in the new processes, methodology, and tools provide advantages to organizations such as:

- Shared knowledge of the aim of the project aligns and motivates the people involved in the project.
- The knowledge shared within the organization about what the stakeholder needs are and the conditions of use aligns the development, introduction and sales efforts.
- A shared vision of the relationship between stakeholder needs and technical characteristics eliminate the contradictions in the development of the system.
- A multitude of alternative solutions at early stages of development, followed by rigorous selection, reduces the risk of developing the first idea and pushing into a dead end
- Early recognition of potential manufacturing challenges and difficulties reduces the risk of encountering bad surprises late in the process

In our case, the documents of DREAM QMS will make up the valuable educational package recommended by Bylund, Wolf, & Mazur (2009).

As stated before, a collaborative working environment is suggested (De Weck & Willcox, 2010), where meetings can take place in a physical and/or virtual space. It is important that, at the beginning of the project and in order to lead the project, the management defines who is going to play the role of Project Manager; this role will maintain order and achieve the organizational objectives.

Organizations that apply the methodology in a systematic and consistent manner, little by little and over time, will advance faster and faster in their learning curve and improve in practice, making the time spent on these processes more efficient and consequently reducing costs (Niebel and Freiwalds, 2009).

6.5 Conclusion

This chapter presented *DREAM*, the proposed quality assurance methodology for system requirements definition; it includes the processes of translating stakeholder needs into stakeholder requirements and translating stakeholder requirements into system requirements.

Moreover, it was presented *DREAM* as a quality management system, the tool that provides support to DREAM methodology.

However, analysis and synthesis is purely intellectual creation, it is widely recommended to have an automatized tool to support requirement management. DREAM as a QMS is totally manual.

DREAM may be de basis for a future software development that meets current industry needs in term of requirement management. Let's remember the desired automated tool features (Ryan & Faulconbridge, 2016; Cuiller, 2015):

- Provide support during the elicitation of stakeholder requirements
- Allow readers to explore, retrieve specific requirements to be retrieved, and generate requirements reports from specific search criteria

- Support the generation of requirements that meet the characteristics of a good requirement
- Allow export and import of requirements in various formats such as word processors or spreadsheets
- Support in change control and related evaluation
- Support functional localization and functional-to-physical translation
- Does not impose any requirements engineering process
- Store and retrieve requirement documents from all levels of development
- Edit and store requirements with their appropriate attributes
- Support requirements validation and verification
- Establish traceability links (forward and backward) between requirements, as well as between requirements and validation and verification elements

Chapter 7 will present DREAM verification against ISO 15288 (2015), and DREAM validation through several case studies.

CHAPTER 7 DREAM Verification & Validation

7.1 Introduction

This chapter addresses the verification and validation of the proposed methodology. The verification is done through: a) DREAM as a QMS compliance with ISO 15288 (2015) for quality assurance purposes; b) DREAM as a QMS compliance with *lean enablers* (Oehmen, 2012) to reduce waste and add value to the processes; and c) DREAM as a QMS compliance with some *desired quality characteristics of Requirements Engineering models* (Batra & Bhatnagar, 2017). Due to the nature of the research (Saunders, Lewis, & Thornhill, 2012), the validation is performed through two qualitative methodologies:

a. *Case Study:* the application of DREAM as a QMS allowed us:
   i. to conceptualize systems which validate that the stakeholder needs were understood, and later, were translated into system requirements; and
   ii. to generate documentation (records) that demonstrate the methodology's follow-up, step by step
b. *Questionnaires* to obtain feedback for evaluating the proposal and subsequently improve it.

Moreover, this chapter presents the case studies, the results from these experiments, and the information obtained from the questionnaires to identify any opportunities for improvement.

7.2 DREAM Verification

DREAM verification will be conducted like an internal *quality audit*[10], in which every DREAM document (processes, work instructions, and forms) will be evaluated.

According to Jones, Ross, & Ruusalepp (2009), "*auditors need to be able to gather evidence, so an inquisitive nature and ability to quickly understand systems to determine where information is likely to be found are essential*." In our case, the audit will be conducted by only one researcher part of our team, who is a certified internal quality auditor.

The documental audit focuses on the identification of *how* the documents comply with the standard. The auditor starts asking questions based on *What? How? When? How much? Who?*

[10] Quality audit: a quality audit is a structured, independent process to determine if project activities comply with organizational and project policies, processes, and procedures [A *Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition]* (ISO 24765, 2017, p. 361)

and *Why?* every task and/or activity demanded by the standard should be documented, then performed. For example, for the requested outcome "*Stakeholders of the system are identified*", the auditor can ask:

- *What* document describes the outcome "*Stakeholders of the system are identified*"?
- Does the document describe *when* the stakeholders of the system are identified?
- Does the document describe *how* the stakeholders of the system are identified?
- Does the document describe *who* identify the stakeholders?
- Does the document describe *why* the stakeholders of the system are identified?

As it has been said, these questions are examples, an internal auditor might ask a lot of related questions in order to understand how the process is supposed to comply with the standard. Nevertheless, what happens if the documents to describe a process that indeed complies with the standard *but* the process is a *wrong* (not well designed, nor efficient) process? To help to prevent that this potential problem is present in DREAM methodology, and as a part of our general objective, we will verify that DREAM includes the good practices of *lean* suggested by Oehmen (2012).

In the following sub-sections, DREAM as a QMS will be evaluated taking into account: a) the standard ISO 15288 (2015), b) the lean enablers (Oehmen, 2012), and c) some desired quality characteristics of Requirements Engineering models (Batra & Bhatnagar, 2017).

DREAM as a QMS compliance with ISO 15288 (2015)

DREAM as a QMS was verified against the two following ISO 1588 (2015) technical processes:

a) 6.4.2 Stakeholder needs and requirements definition process, and
b) 6.4.3 System requirements definition process

For each technical process, ISO 15288 (2015) contains sections such as *Purpose*, *Outcomes*, and *Activities and tasks*. In this sense, the following sub-sections present the *Purpose* of the technical processes; in contrast, the *Outcomes* and *Activities and tasks* will be compared against the DREAM as a quality management system (QMS) (processes, work instructions, and forms) to verify that the DREAM methodology meets the ISO 15288 (2015) requirements.

Technical Process 6.4.2 Stakeholder needs and requirements definition process

"**6.4.2.1 Purpose:** *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. It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs. It analyzes and transforms these needs into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that is the reference against which each resulting operational capability is validated. The stakeholder requirements are defined considering the context of the system of interest with the interoperating systems and enabling systems."* (ISO 15288, 2015)

Table 66 shows the *6.4.2.2 Outcomes* (ISO 15288, 2015) versus *DREAM as a QMS*. The first column lists the expected outcomes for the technical process; in the second column, the percentage of ISO 15288 (2015) fulfillment; and, the third column lists the DREAM documents through which is fulfilled the expected outcome.

Table 66 ISO 15288 (2015) 6.4.2.2 Outcomes versus DREAM as QMS

6.4.2.2 Outcomes (ISO 15288, 2015)	DREAM as QMS	DREAM Documents
a) Stakeholders of the system are identified.	100%	P-1, W-1, F-1a, F-1b
b) Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.	100%	P-1, W-2, F-2a, F-2b, F-2c
c) Constraints on a system are identified.	100%	P-1, W-3, F-3a, F-3b, F-3c
d) Stakeholder needs are defined.	100%	P-1, W-2, F-2a, F-2b, F-2c W-3, F-3a, F-3b, F-3c
e) Stakeholder needs are prioritized and transformed into clearly defined stakeholder requirements.	100%	P-1, W-5, F-5 W-6, F-6
f) Critical performance measures are defined.	100%	P-1, W-7, F-7a
g) Stakeholder agreement that their needs and expectations are reflected adequately in the requirements is achieved.	100%	P-1, W-3, F-3c W-4, F-4a, F-4b W-7, F-7a, F-7b
h) Any enabling systems or services needed for stakeholder needs and requirements are available.	100%	P-1, the set of work instructions (W) and forms (F)
i) Traceability of stakeholder requirements to stakeholders and their needs is established.	100%	P-1, the set of work instructions (W) and forms (F)

Table 67 shows ISO 15288 (2015) *6.4.2.3 Activities and tasks* versus DREAM as QMS. The first column lists activities and tasks for the technical process; in the second column the percentage of ISO 15288 (2015) fulfillment, and the DREAM documents through which is fulfilled the expected tasks.

Note: this table contains listed only the activities and their tasks; the reader will find in Annexes the standard ISO 15288 (2015) *6.4.2.3 Activities and tasks* including the Notes for each activity and/or task if they exist.

Table 67 ISO 15288 (2015) 6.4.2.3 Activities and tasks versus DREAM as QMS

6.4.2.3 Activities and tasks (ISO 15288, 2015)	DREAM as QMS
a) Prepare for Stakeholder Needs and Requirements Definition. This activity consists of the following tasks:	100%
1) Identify the stakeholders who have an interest in the system throughout its life cycle.	P-1, W-1, F-1a, F-1b
2) Define the stakeholder needs and requirements definition strategy.	P-1, W-2, F-2a, F-2b, F-2c, W-3, F-3a, F-3b, F-3c W-4, F-4a, F-4b
3) Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition.	P-1, the set of work instructions (W) and Forms (F)
4) Obtain or acquire access to enabling systems or services to be used.	P-1, the set of work instructions (W) and forms (F)
b) Define stakeholder needs. This activity consists of the following tasks:	100%
1) Define the context of use within the concept of operations and the preliminary life cycle concepts.	P-1, W-2, F-2a, F-2b, F-2c
2) Identify stakeholder needs.	P-1, W-3, F-3a, F-3b, F-3c, W-4, F-4a, F-4b
3) Prioritize and down-select needs.	P-1, W-5, F-5
4) Define the stakeholder needs and rationale.	P-1, W-3, F-3a, F-3b, F-3c

c) Develop the operational concept and other life cycle concepts. This activity consists of the following tasks: *1) Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other life cycle concepts.* *2) Identify the interaction between users and the system.*	100% P-1, W-2, F-2a, F-2b, F-2c, W-3, F-3a, F-3b, F-3c P-1, W-3, F-3a, F-3b, F-3c
d) Transform stakeholder needs into stakeholder requirements. This activity consists of the following tasks: *1) Identify the constraints on a system solution.* *2) Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health.* *3) Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics.*	100% P-1, W-3, F-3a, F-3b, F-3c, W-4, F-4a, F-4b P-1, W-3, F-3a, F-3b, F-3c P-1, W-5, F-5, W-6, F-6, W-7, F-7a, F-7b
e) Analyze stakeholder requirements. This activity consists of the following tasks: *1) Analyze the complete set of stakeholder requirements.* *2) Define critical performance measures that enable the assessment of technical achievement.* *3) Feedback the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed.* *4) Resolve stakeholder requirements issues.*	100% P-1, W-7, F-7a, F-7b P-1, W-7, F-7a, F-7b P-1, W-7, F-7a, F-7b, W-7, F-7a, F-7b P-1, W-4, F-4a, F-4b
f) Manage the stakeholder needs and requirements definition. This activity consists of the following tasks: *1) Obtain explicit agreement on the stakeholder requirements.* *2) Maintain traceability of stakeholder needs and requirements.* *3) Provide key information items that have been selected for baselines.*	100% P-1, W-4, F-4a, F-4b, W-7, F-7a, F-7b P-1, the set of work instructions (W) and forms (F) P-1, the set of work instructions (W) and forms (F)

Technical Process 6.4.3 System requirements definition process

"**6.4.3.1 *Purpose:*** *The purpose of the System 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. This process creates a set of measurable system requirements that specify, from the supplier's perspective, what characteristics, attributes, and functional and performance requirements the system is to possess, in order to satisfy stakeholder requirements. As far as constraints permit, the requirements should not imply any specific implementation."* (ISO 15288, 2015)

Table 68 shows the *6.4.3.2 Outcomes* (ISO 15288, 2015) versus *DREAM as QMS*. The first column lists the expected outcomes for the technical process; in the second column, the percentage of ISO 15288 (2015) fulfillment; and, the third column lists the DREAM documents through which is fulfilled the expected outcome.

Table 68 ISO 15288 (2015) 6.4.3.2 Outcomes versus DREAM as QMS

6.4.3.2 Outcomes (ISO 15288, 2015)	DREAM as QMS	DREAM Documents
a) The system description, including system interfaces, functions and boundaries, for a system solution is defined.	100%	P-1, W-2, F-2a, F-2b, F-2c
b) System requirements (functional, performance, process, non-functional, and interface) and design constraints are defined.	100%	P-2, W-9, F-9, W-11-F-11a, F-11b
c) Critical performance measures are defined.	100%	P-2, W-11-F11-a, F-11b
d) The system requirements are analyzed.	100%	P-2, W-8, F-8
e) Any enabling systems or services needed for system requirements definition are available.	100%	P-2, the set of work instructions (W) and forms (F)
f) Traceability of system requirements to stakeholder requirements is developed.	100%	P-2, W-9, F-9, W-10, F-10a, F-10b

Table 69 shows ISO 15288 (2015) *6.4.3.3 Activities and tasks* versus DREAM as QMS. The first column lists activities and tasks for the technical process; in the second column the percentage of ISO 15288 (2015) fulfillment, and the DREAM documents through which is fulfilled the expected tasks.

Note: this table contains listed only the activities and their tasks; the reader will find in Annexes the standard ISO 15288 (2015) *6.4.3.3 Activities and tasks* including the Notes for each activity and/or task if they exist.

Table 69 ISO 15288 (2015) 6.4.3.3 Activities and tasks versus DREAM as QMS

6.4.3.3 Activities and tasks (ISO 15288, 2015)	DREAM as QMS
a) Prepare for System Requirements Definition. This activity consists of the following tasks:	100%
1) Define the functional boundary of the system in terms of the behavior and properties to be provided.	P-2, W-8, F-8
2) Define the system requirements definition strategy.	P-2, W-9, F-9
3) Identify and plan for the necessary enabling systems or services needed to support system requirements definition.	P-2, W-9, F-9
4) Obtain or acquire access to enabling systems or services to be used.	P-2, W-9, F-9
b) Define system requirements. This activity consists of the following tasks:	100%
1) Define each function that the system is required to perform.	P-2, W-9, F-9
2) Define the necessary implementation constraints.	P-2, W-9, F-9
3) Identify system requirements that relate to risks, the criticality of the system, or critical quality characteristics.	P-2, W-9, F-9
4) Define system requirements and rationale.	P-2, W-9, F-9
c) Analyze system requirements. This activity consists of the following tasks:	100%
1) Analyze the complete set of system requirements.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F-11b
2) Define critical performance measures that enable the assessment of technical achievement.	P-2, W-11-F-11a
3) Feedback the analyzed requirements to applicable stakeholders for review.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F11b
4) Resolve system requirements issues.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F11b
d) Manage system requirements. This activity consists of the following tasks:	100%
1) Obtain explicit agreement on the system requirements.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F11b
2) Maintain traceability of the system requirements.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F11b
3) Provide key information items that have been selected for baselines.	P-2, W-9, F-9, W-10, F-10a, F-10b, W-11, F-11a, F11b

DREAM as a QMS compliance with lean enablers

The following Table 70 shows the lean enables proposed by Oehmen (2012, pp. 177-178) and how DREAM as QMS meets the lean enabler for Managing Engineering Programs.

Table 70 Lean enablers for Stakeholder Requirements Definition Process (Oehmen, 2012) versus DREAM as a QMS

SE Process	Lean Enabler #	Lean Enabler for Managing Engineering Programs	How DREAM as QMS meets the lean enabler for Managing Engineering Programs
4		**System Engineering: Technical Processes**	
4.1		**Stakeholder Requirements Definition Process**	
4.1	2.1	Establish the value and benefit of the program to the stakeholders	Providing timely information on the benefits to be gained from implementing the program, it should be included by management during the implementation of the methodology.

4.1	2.1.1	Define value as the outcome of an activity that satisfies at least three conditions. A. The external customer stakeholders are willing to pay for value. B. Transforms information or material or reduces uncertainty. C. Provides specified programs benefits right at the first time	The definition of "value" is made known in terms of a) stakeholders paying for what they themselves consider "valuable" (W-2, F-2a, F-2b, F-2c, W-5, F-5); b) having the right information and in a timely manner will reduce project uncertainty (the set of work instructions W and forms F); c) providing the right benefits from the first time (quality) (W-3, F-3a, F-3b, F-3c)
4.1	2.1.2	Define value – added in terms of the value of the customer stakeholders and their needs	By conducting the on-site visit and getting to know the culture of the organization, identifying the needs of the stakeholders (W-3, F-3a, F-3b, F-3c) and prioritizing them (W-5, F-5), the analysis and design team knows what is "valuable" for the stakeholders (W-2, F-2a, F-2b, F-2c) and will work to develop a system that meets those expectations.
4.1	2.1.3	Develop a robust process to capture, develop, and disseminate customer stakeholder value with extreme clarity	The proposed forms will support the capture, development, and dissemination of customer stakeholder value with extreme clarity (W-2, W-2, W-5, F-5)
4.1	2.1.4	Proactively resolve potential conflict stakeholder values and expectations, and seek consensus	Through the verification of feasibility (W-4, F-4a, F-4b), potential conflicts between stakeholder needs will be resolved, first by individually identifying those that are "not feasibility" and then by identifying those that oppose or contradict each other.
4.1	2.1.5	Explain customer stakeholder culture to Program employees, i.e. the value system, approach, attitude, expectations, and issues	The methodology proposes to develop collaborative sessions in the stakeholder facilities (W-2, F-2a, F-2b, F-2c, W-3, F-3a, F-3b, F-3c); the objective is that the analysis and design team may be able to observe the stakeholder culture, identify the unspoken needs, and identify where the “value” is for the stakeholders. Through the "visit to the crime scene" or "gemba" the culture of the stakeholders is known
4.1	2.3.10	Clearly track assumptions and environmental conditions that influence stakeholder requirements and their perception of program benefits	The development of collaborative sessions in the stakeholder facilities (W-2, F-2a, F-2b, F-2c, W-3, F-3a, F-3b, F-3c) will allow the identification of environmental conditions and identify the assumptions through direct observation and questionnaires; the objective is that the analysis and design team identify what is valuable for the stakeholders.
4.1	2.5	Clarify, derive and prioritize requirements early, often and proactively	By applying the methodology (W-3, F-3a, F-3b, F-3c, W-4, F-4a, F-4b, W-5, F-5, W-6, F-6) the requirements may be clarified, derived, and prioritized in an early, often, and proactively manner.
4.1	2.5.4	Listen for and capture unspoken customer requirements	The methodology proposes to do “gemba” to know the stakeholder culture and easy the task of capturing the unspoken customer requirements. Also, the methodology includes the application of

			the questionnaire (W-3, F-3a, F-3b, F-3c) to identify, as far as possible, the completeness stakeholder needs. Finally, through the definition of Decomposed and Derived requirements, activities contemplated in the proposed methodology and to be performed by the analysis and design team (W-6, F-6).
4.1	2.5.6	Actively promote the maturation of stakeholder requirements, e.g., by providing detailed trade-off studies, feasibility studies, and virtual prototypes	The promotion of stakeholder maturity requirements will be achieved through the feasibility analysis proposed in the methodology (W-4, F-4a, F-4b); trade-off studies and prototyping, two techniques used to validate that a system will truly meet their expectations, are beyond this research work boundaries.
4.1	2.5.7	Facilitate communication between different and possibly diverging stakeholders to develop a shared understanding of the program among the stakeholders, clearly identifying and incorporating the various interest of different stakeholders (aligned, indifferent, or opposed), and establish trust	Through the early identification of the stakeholders involved in each requirement, activity incorporated in the proposed methodology (W-1, F-1a, F-1b); the communication will be facilitated through the support tool that will allow the collaborative work via remote in real time and can interact through video, chat and sharing a design screen.
4.1	2.5.8	Create effective channels for clarification of requirements (e.g., involving customer stakeholders in program teams)	Through the early identification of the stakeholders involved in each requirement, activity incorporated in the proposed methodology (W-1, F-1a, F-1b); the communication will be facilitated through the support tool that will allow the collaborative work via remote in real time and can interact through video, chat and sharing a design screen.
4.1	2.5.10	Employ agile methods to manage necessary requirements change and make the program deliverables robust against those changes. Make both program processes and program deliverables reusable, reconfigurable, and scalable	The proposed support tool would allow the application of agile methods during requirement change management. This tool would allow the collaboration in real time of the work team achieving agility; the involved stakeholders could be in contact via remote and could interact through video, chat, and sharing a design screen if required. The data stored in the databases could be reusable.
4.1	3.5.14	The program manager must personally understand, clarify and remove ambiguity, conflicts, and waste from key requirements and expectations al the program start	In the early stages of detecting and writing stakeholder needs, it is verified that there is no opposition among them (W-4, F-4a, F-4b), which avoids working on a need that cannot be met and would cause waste of resources; later on, when translating these needs into stakeholder requirements, the quality characteristics they must comply with are verified, among them feasibility, ambiguity, consistent, to eliminate those that cannot be incorporated and not work on them, thus avoiding the waste of resources (W-7, F-7a, F-7b).
4.2		**Requirements Analysis Process**	

4.2	3.10.9	Perform robust system architecting and requirement analysis to determine technology needs and current technology readiness levels	Through the review of the quality characteristic of the requirements, specifically: feasible; the requirements must be verified in a timely manner as to whether the means and technology available for its manufacture exist, this is an activity included in the proposed methodology (W-8, F-8, W-9, F-9, W-10, F-10, W-11, F-11a, F11b)

DREAM as a QMS compliance with desired quality characteristics of Requirements Engineering models

The following Table 71 shows some desired quality characteristics of Requirements Engineering models (Batra & Bhatnagar, 2017) and how DREAM as QMS meets them.

Table 71 Desired quality characteristics of Requirements Engineering models (Batra & Bhatnagar, 2017) versus DREAM as a QMS

Desired quality characteristics of Requirements Engineering models (Batra & Bhatnagar, 2017)	DREAM as QMS	How DREAM as QMS meets the desired model quality characteristic
Linearity	100%	P-1, P-2, and the set of work instructions (W) are linear, the processes were designed with the premise that activities and task could be performed following coherent order
Support for changing requirements	100%	P-1, P-2, and the set of work instructions (W) support the evolution of needs and/or requirements.
Interactive in nature	100%	P-1, P-2, and the set of work instructions (W) are interactive in nature, as they were designed to be applied in a collaborative environment where interaction among stakeholders is possible
User feedback	100%	P-1, P-2, and the set of work instructions (W) allows stakeholder participation, included user feedback
Support for reverse engineering	100%	P-1, specifically in W-3, allows to apply reverse engineering since the elicitation of stakeholder needs is proposed to be done through a questionnaire based on a system requirement classification
Risk assessment	100%	P-1 and P-2 allows risk assessment; W-2 starts considering uncertainty, while W-7 and W-11 allow the risk identification, the risk management, and the risk mitigation of each stakeholder (W-7) or system (W-11) requirement.
Criteria for application specific requirements elicitation technique	100%	P-1 specifically in W-3 proposes a specific requirement elicitation technique: a questionnaire based on a system requirement classification.
Requirements pre-processing	100%	P-1, in the W-4, allows the analysis of stakeholder needs, previous to their translation into stakeholder requirements; W-4 allows the identification of contradictions and stakeholder negotiation to mitigate them.
Requirements prioritization	100%	P-1, specifically through W-5 allows to prioritize stakeholder needs, and with aid of the unique identifier of each need, when needs are translated into requirements, they are already prioritized maintaining at the same time their traceability.
Effort estimation	0%	DREAM does not allow to estimate the effort of applying it in a project

7.3 DREAM Validation: Case studies and Results

The case studies were led in the Tecnológico Nacional de México, Instituto Tecnológico de Toluca (ITTol), Department of Mechatronic Engineering, with the collaboration of Professor Citlalih A. Y. Gutierrez Estrada, in student projects. The selected projects possessed some specific characteristics: a) they had to be completed in one semester, b) if possible, being a real case study to be conducted inside a small enterprise, c) they should allow the conceptualization of a system and the development of a first prototype, d) involving students with no experience in system conceptualization, e) each student team should be of different domains, like mechanical, electrical, or software engineers. These conditions allowed us to test DREAM in a short period, several times in parallel, in different domains, validate the prototype with real stakeholders and identify improvement opportunities for DREAM as a QMS. Professors played the role of a *Project manager*, and students played the role of *analysis and design team*. The case studies are presented below.

The case studies were developed in two different advancement stages of this research work:

1. When the process *from stakeholder needs to stakeholder requirements* was designed, it was applied to only one case study: embedded system with connected modules “Pill-Bot”
2. When the process *from stakeholder requirements to system requirements* was designed, both processes were applied to three case studies:
    - Information system to monitor stoppages in production lines;
    - Web and mobile device information system for online shopping; and
    - Wireless loader by magnetic induction through an electrical generator.

Following sub-sections present the case studies and their results.

Case study: embedded system with connected modules “Pill-Bot”

Students of Mechatronic Engineering from the ITTol detected that there is a need for patients to effectively follow medical treatments that require the programmed intake of drugs. With the support of their teachers, the students developed a "programmable system for dispensing drugs for dependents through controllers and blue-tooth connection" named Pill-Bot, in other words, an intelligent pill dispenser, whose first prototype can be seen in Figure 84.

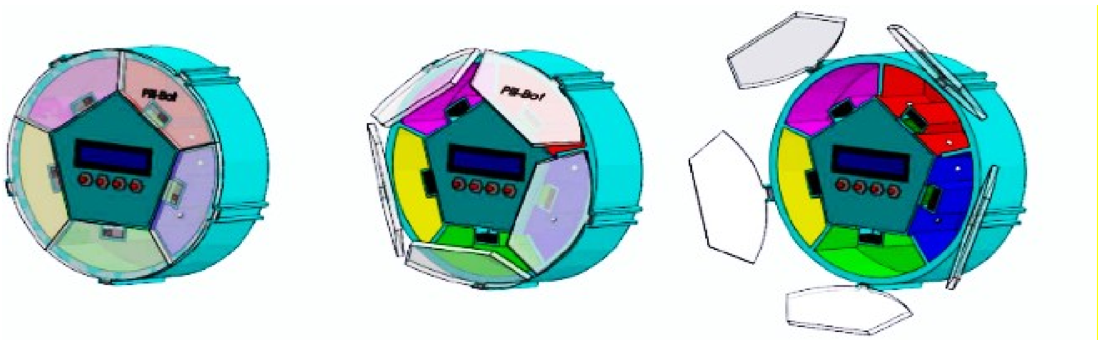


Figure 84 Virtual prototype of the intelligent pill dispenser in Solidworks.

Results

This case study made possible to evaluate the first expected result of this research work:

1. A methodology and its tools to ensure, as far as possible, that *the analysis and design team has understood the stakeholder needs and has translated them into system requirements.*

This first validation was done through the evaluation of the prototype; in other words, the prototype was used to validate if the analysis and design team has understood the stakeholder needs and translated them into system requirements, resulting in a system that fulfills the stakeholder needs.

"Conduct validation to demonstrate conformance of services to stakeholder requirements… Requirements validation is subject to approval by the project authority and key stakeholders. This process is invoked during the Stakeholders Requirements Definition Process to confirm that the requirements properly reflect the stakeholder needs and to establish validation criteria, i.e. that we have the right requirements. System validation confirms that the system, as built, satisfies stakeholder stated needs and requirements, that it is the right system" (ISO 29148, 2011, p.36)

This prototype evaluation was done through a questionnaire answered by sixteen stakeholders who played the role of *users*. Figure 85 illustrates the questionnaire that evaluated user needs as maintainability, portability, reliability, efficiency, usability, and functionality.

INSTITUTO TECNOLÓGICO DE TOLUCA

Formato para la Evaluación de Prototipos de Sw (Stakeholders)

Título del Proyecto: ***SISTEMA DOSIFICADOR DE MEDICAMENTOS PARA PERSONAS DEPENDIENTES MEDIANTE MICROCONTROLADORES Y CONEXIÓN BLUETOOTH***

Folio: 12

Elaboró: Rodrigo Hernández Barrera, Ricardo Merino Leyva, Arturo Morales Cuadros y Dra. Citlalih Gutiérrez E.

Fecha: 27/07/2017

Objetivo: Evaluar las características del **primer prototipo del dispositivo PILLBOT.**

Instrucciones para el Evaluador:
- Después de haber sido informado acerca de las características del primer el prototipo y su funcionamiento, contesta las preguntas.
- Coloca X en la respuesta de tu elección.

		SI	NO
Funcionalidad	1. ¿Los colores usados en el prototipo resultan atractivos para usted?	X	
	2. ¿Considera que el brazalete que se enlazo con el dispositivo es útil para alertar al usuario cuando este está lejos de la ubicación del dispositivo?	X	
	3. ¿Los tipo de alarmas implementadas en el dispositivo resultan útiles para alertarlo del horario de medicación?	X	
	4. ¿En su opinión el método de registro usado para el dispositivo es suficiente para dar seguimiento al usuario?		X
Usabilidad	5. ¿Después de leer las instrucciones, comprendió el método para programar las alarmas en el dispositivo?	X	
	6. ¿El periodo de tiempo durante el cual las alarmas están activadas es suficiente para que el usuario ingiera la medicación?	X	
	7. ¿La fuente usada en la pantalla del dispositivo fue lo suficientemente grande para que la pudiera leer sin problemas?	x	
Eficiencia	8. ¿Las dimensiones del dispositivos son los suficientemente pequeñas?		x
	9. ¿La capacidad de almacenamiento del dispositivo es suficiente para un tratamiento mensual?	x	
	10.¿Considera que el número de compartimentos para medicamentos en el dispositivo es suficiente para un tratamiento médico?	x	
Confiabilidad	11. ¿Cree que el diseño de los compartimentos del dispositivo sea capaz de proteger a los medicamentos?	x	
	12. ¿Visualizo alguna falla en el funcionamiento del dispositivo?		X
	13. Si la respuesta a la pregunta anterior fue afirmativa, ¿Cree que la falla del dispositivo pueda interferir con el tratamiento médico?		
Portabilidad	14. ¿En su opinión el dispositivo posee la capacidad de ser portable, es decir, es capaz de transportarlo de un lugar a otro?		X
	15. Si la respuesta a la pregunta anterior fue afirmativa, ¿Considera que el dispositivo es capaz de resistir movimientos repentinos sin dañar el medicamento?		
	16.¿Considera que el puerto de entrada para el cable de alimentación es un puerto común?	X	
	17. ¿Visualiza algún problema que pueda ocurrir al ingresar medicamento al dispositivo actual?		X
Mantenibilidad	18. ¿Considera que la información enviada al brazalete que acompaña al dispositivo es suficiente para dar seguimiento al paciente?	X	
	19. ¿Cree que el manual que acompaña al dispositivo es suficiente para reiniciar o reparar el mismo en caso de que este falle?	X	

Observaciones o Recomendaciones:

Nombre del evaluador: Guillermo Ortiz

Nombre del evaluador:

Figure 85 Record of the questionnaire to evaluate the Pill-Bot prototype

Figure 86 presents the results of the prototype evaluation; in general, *users* consider that the Pill-bot prototype met their expectations regarding to its maintainability, reliability, efficiency, usability, and functionality (between 71 and 97% satisfaction); in contrast, the prototype can improve its portability (55%) to increase user satisfaction.

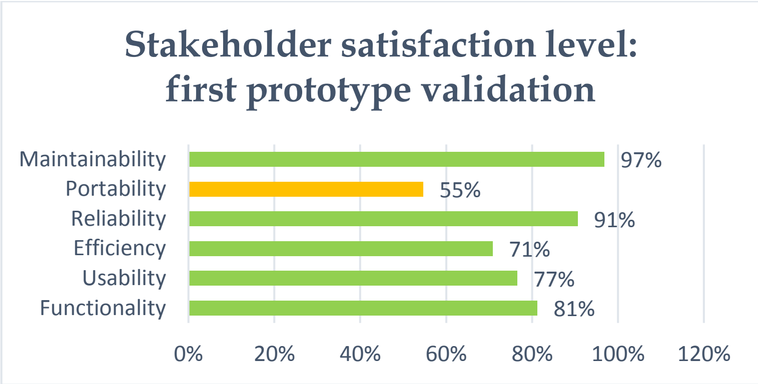


Figure 86 Results of Pill-bot prototype validation

Case study: Information system to monitor stoppages in production lines

Industry "X" (the company name is not mentioned for confidential reasons) is implanted in six countries in Latin America; it manufactures several kinds of packaging like industrial and food service packaging, disposable medical supplies, and packaging for mass consumption. After conducting a diagnostic in industry "X", students from the Computing Engineering Department realized that one of the reasons for economic losses was the uncontrolled stoppage in production lines. They proposed to design an information system to monitor these stoppages in order to measure, control, and improve the situation. Figure 87 illustrates how is conceptualized the information system: first, the sensor detects line stops; later, data is collected at the station to send notifications to personnel (e-mail or SMS message); finally, data is sent to the cloud to display the line stop on the screen (real time).

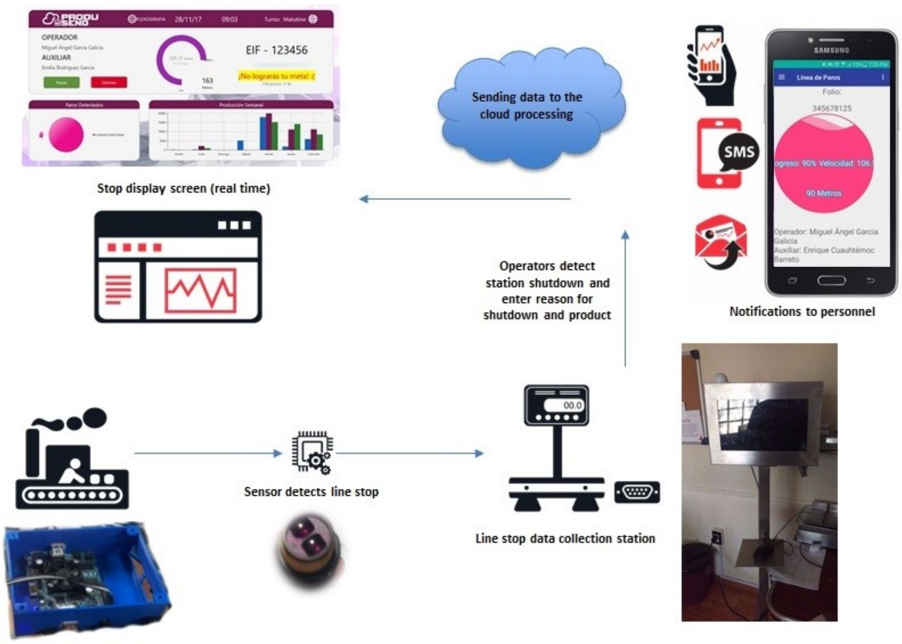


Figure 87 Information system for monitoring stops in a production line

Case study: web and mobile device information system for online shopping

On-line shopping is a practical way to save time, but it is not always easy to compare the prices of products. Students carried out a market study to find out the opinion of consumers when buying clothes online. Their study revealed that 88% of buyers would like to have an application to accede easier to several cloth stores and compare product prices.

In this case study, students designed a web and mobile device information system for online shopping. Figure 88 illustrates the different screens: for entering or register, identify oneself, select clothes of men, women, or children, and finally, options of shopping for clothes.

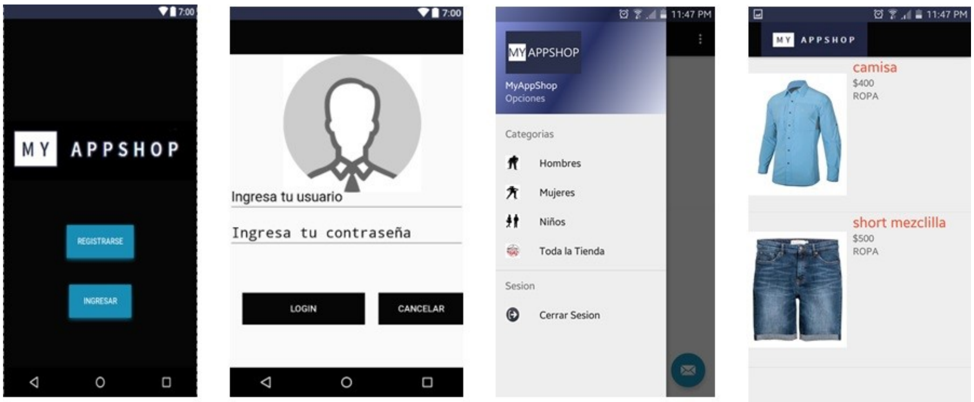


Figure 88 Screen examples of the Web application on a mobile phone for online shopping

Case study: wireless loader by magnetic induction through an electrical generator

It is fundamental to have electricity to do daily activities; the diverse electrical devices need an electrical feeding source and conductors to transfer the energy. ITTol Mechanical and Electric Engineering students observed that people who use bicycles as mean of transportation could generate the energy to charge their own mobile devices; nevertheless, one current problem to face is that wires make the solution non-practical and unsightly. The students proposed to design a system (electrical generator) that, assembled in the rims of a bicycle, would produce the amount of energy needed to load a mobile device in a wireless way through the application of magnetic fields. To this goal, they designed a module that reacts as a Wi-Fi network, in order to wirelessly transfer the electric power to the mobile device. Figure 89 illustrates the wireless electrical generator assembled to the rims of the bike.

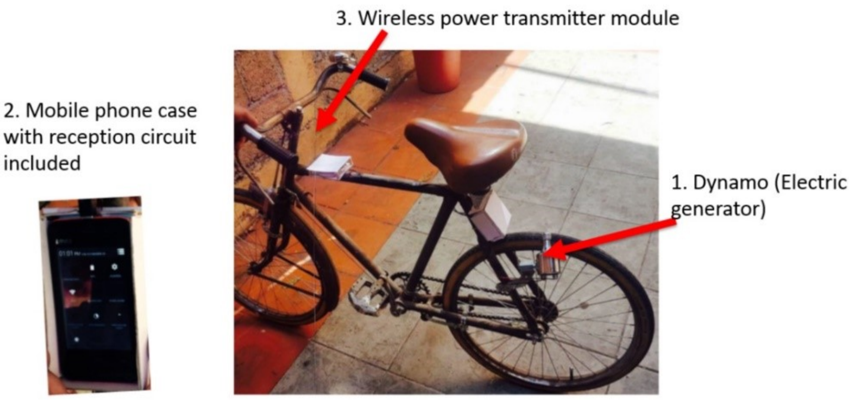


Figure 89 Wireless electrical generator

Results

In these case studies it was possible to evaluate the second expected result of this research work:

> *2. The validation of the proposition through a case study (or more), generating documentation that demonstrates the methodology's follow-up, step by step, from the translation of stakeholder needs into system requirements.*

This validation was conducted through the evaluation of the methodology and its tools. This validation was done through a questionnaire that was answered by the thirteen students involved in the case studies, who played the role of *analysis and design team*.

Following the recommendations of Tapia Moreno (2010), we designed the questionnaire in three steps: 1) design the draft, define what aspects we are trying to cover, and how the elements will be distributed in the different content areas; 2) decide on the format of the questions to be used, their general distribution, the length of the questionnaire (number of questions), how it will be applied, when it will be carried out, instructions, and the scoring system we are going to use, which will depend on the type of questions; and 3) identify an erroneous formulation of the question, and subsequently rewrite the question.

In our case, we designed the questionnaire as follows:

1) The aspects we tried to cover were: first, if DREAM provided support when translating the stakeholder needs into system requirements, if DREAM was coherent, and included the required activities; second, if DREAM work instructions were clear and easy to follow; and third, if DREAM forms were easy to fill and ergonomically suitable (font size, color, field size, number of fields provided); we decided to distribute the areas of the questionnaire following this order: methodology, work instructions, forms, and overall experience; the questions were grouped in the same sense.
2) To gather punctual information, the questionnaire had closed-ended dichotomous questions (yes/no as possible answers); moreover, the questionnaire included open-ended questions that allowed students to express with complete freedom about what they thought, make suggestions to improve, and describe their experience after applying DREAM; closed-ended and open-ended questions were interspersed to avoid respondent fatigue; we decided to include twenty questions; finally, to assure that we get truthful answers, the questionnaire would be answered by the involved students after the project was finished, and their notes were already assigned by professors.
3) We verified that for each question only one answer could be possible, the content of the question must refer only to the object of our study and the specific variable we want to investigate, the questions were formulated with precision, taking care that each question does not exceed fifteen words, and being as objective as possible with no suggestion of what is desired as an answer.

To ensure the reliability of the questionnaire, the responses of each team (three last case studies) were grouped together and then compared with each other. It was found that no matter which team they belonged to, the students responded in a very similar way, so we decided to group all the answers to be studied together as one group.

The validity of the questionnaire took into account the a) validity of content, which alludes to the need to ensure that the questionnaire constitutes an adequate and representative sample of the variables to be evaluated with it; and b) consensual validity, since we, as a research team, consider that the questionnaire measures what it is intended to measure (Tapia Moreno, 2010).

The students' answers allowed us to identify separately the strengths and/or weakness of the methodology, the work instructions, and forms, as well as opportunities for improvement. Figure 90 illustrates the questionnaire that evaluates the methodology and its tools (work instructions and forms).

Formato para la de evaluación de la metodología propuesta y sus herramientas

Trabajo de investigación: Metodología para la transformación de necesidades en requisitos integrando Ingeniería de Sistemas, Calidad y Pensamiento Esbelto
Objetivo: evaluar la metodología propuesta y sus herramientas
Instrucciones para el evaluador:

- Después de haber aplicado la metodología y sus herramientas en el caso de estudio, conteste las preguntas
- Coloque X en la respuesta de su elección y agregue sus comentarios en caso de existir

Nombre del evaluador: Lesly Nohemi Cruz Gomez **Fecha:** 07/06/2018 **Folio:** 3

	Preguntas	Si	No
Metodología	1. Según su experiencia desarrollando nuevos sistemas, ¿es coherente el orden de las etapas y actividades de la metodología propuesta?	X	
	2. En caso de haber respondido negativamente, ¿podría sugerir el orden correcto? No aplica		
	3. ¿Considera que hizo falta alguna actividad en la metodología propuesta?		X
	4. En caso de haber respondido afirmativamente, ¿cuál(es)? No aplica		
	5. ¿Detectó algún problema al aplicar la metodología?		X
	6. En caso de haber respondido afirmativamente, ¿cuál(es)? Ninguna		
Instrucciones de trabajo	7. ¿Las instrucciones de trabajo fueron lo suficientemente claras?	X	
	8. ¿Las instrucciones de trabajo fueron fáciles de seguir?	X	
	9. ¿Es algún instructivo en particular más complicado que otros de aplicar?		X
	10. En caso de haber respondido afirmativamente, ¿cuál fue? y ¿por qué lo considera así? Ninguna		
	11. ¿Qué sugerencias podría compartir para disminuir o mitigar las dificultades encontradas? Que se explique un poco más cómo llenar estos campos con ejemplos		
Formatos	12. ¿Considera que los campos de los formatos están ordenados conforme se desarrollan las actividades descritas en las instrucciones de trabajo?	X	
	13. ¿Considera que el llenado de los formatos fue fácil?		X
	14. ¿Considera que el tamaño de los campos es el adecuado?	X	
	15. ¿Considera que la cantidad de campos fueron suficientes?	X	
	16. ¿Considera que el tamaño de la letra es el correcto?	X	
	17. ¿Considera que el color de los formatos es agradable a la vista?	X	
	18. En general, describa su experiencia al aplicar la metodología propuesta y sus herramientas. En base a los conocimientos que se nos dieron, pudimos llenar los formatos aunque fue un poco difícil porque está un poco laborioso		
	19. ¿Qué sugerencias podría dar para mejorar la metodología propuesta y sus herramientas? Que se den más ejemplos de formatos previos para poder guiarnos para realizar el llenado de los formatos		
	20. ¿Considera que la metodología logró el objetivo de documentar los requisitos del sistema, agregar valor a los procesos de análisis y diseño, y asegurar la calidad del sistema?	X	

Figure 90 Record of the questionnaire to evaluate DREAM methodology and its tools

Figure 91 shows the results obtained through the questionnaire. The *methodology* was evaluated by questions 1, 2, 3, 4, 5, and 6; the *work instructions* were evaluated by questions 7, 8, 9, 10, and 11; the *forms* were evaluated by questions 12, 13, 14, 15, 16, and 17. Questions 18 and 19 asked for suggestions; question 20 was the *key question* because it evaluated if the methodology achieved, or not, the objective of documenting the system requirements, adding value to the analysis and design processes while assuring the quality of the system. Questions 2, 4, 6, 10, 11, 18, and 19 were open questions, this is the reason why they are not included in the graph; the most representative answers for open questions are shown below.

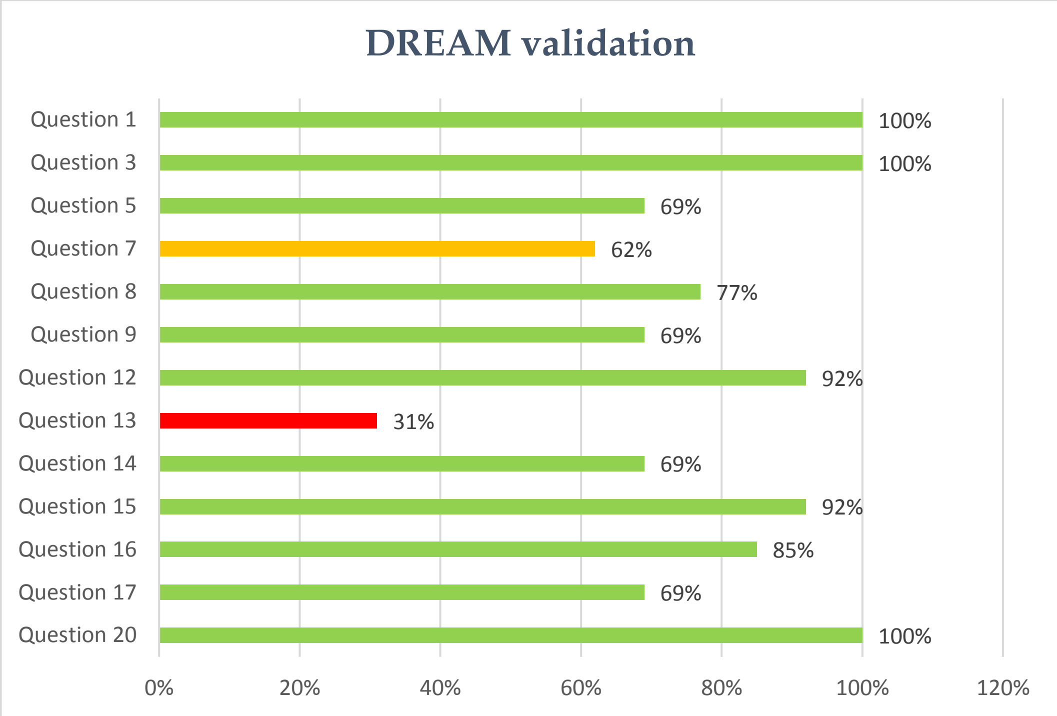


Figure 91 DREAM Validation: questionnaire results

The interpretation of results is detailed here:

Methodology

Question 1: 100% of the students considered that the order of stages and activities of the proposed methodology is consistent

Questions 2: Could you suggest the correct order? Question 2 had no answers due that all the interviewed students considered that the order of the activities is correct.

Question 3: 100% of the students considered that any activity was lacking in the proposed methodology

Questions 4: Which one? Question 4 had no answers due that all the interviewed students considered that there is not any activity missing.

Question 5: 69% of the students did not detect any problem applying the methodology.

Question 6: What problems did you detect when applying the methodology?

- *It takes a long time to perform the activity and sometimes the process becomes tedious*
- *The methodology is very complex for beginners*
- *If there is a lack of communication between team members, the explanation of the system can lead to confusion.*
- *We are not accustomed to documenting the processes we carry out or the systems we develop.*

Work Instructions

Question 7: 62% of the students considered that the work instructions were clear.

Question 8: 77% of the students considered that the work instructions were easy to follow.

Question 9: 69% of the students considered that all the work instructions had, among them, the same difficulty degree to be applied.

Question 10: What work instruction is more complicated to follow compared with the others?

- *The number 3 (elicitation of stakeholder needs)*
- *Some activities were too long and heavy while the others were easy and fast.*
- *Filling out form 5 is very laborious*

Question 11: What suggestions could you share to lessen or mitigate the difficulties encountered?

- *Always try to keep the order of the stages and look for alternatives with other methods*
- *Rely more on new technologies to streamline processes and conduct more consultations with specialists*
- *To explain a little more how to fill in the fields with examples*
- *To establish an effective communication method to use time correctly and efficiently*
- *I suggest a dictionary of words for instructions that contain very technical words*
- *Verify the relevance of all fields in order to reduce work*
- *Regarding the explanation of the instructions, they should be clearer, since I had one idea like some colleagues but others had another, and since we did not fully understand the explanation of the instructions there were some problems.*
- *To establish clearly the link between the forms that guarantee the sequence between them.*

Forms

Question 12: 92% of the students considered that the fields of the forms are ordered as the activities described in the work instructions.

Question 13: 31% of the students considered that filling out the forms was an easy task

Question 14: 69% of the students considered that the size of the fields is adequate.

Question 15: 92% of the students considered that the number of fields was sufficient.

Question 16: 85% of the students considered that the font size is correct.

Question 17: 69% of the students considered that the color of the forms is pleasing to the eye.

General experience using the proposed methodology

Question 18: In general, describe your experience applying the proposed methodology and its tools

- *The experience that we had with the form was pleasant since it took an organization of the components one to one that are taken into account in the project.*
- *It was an experience full of new things, I grew technically by working with these forms, as well as learning words and expressions outside the colloquial.*
- *Personally, I think that at the beginning of the project both the team and I found it a little difficult to apply the methodology because we are not used to working collaboratively, but after each member became involved in each part of the project it was much easier to work.*
- *My experience was good, I learned several things; however, working with the forms was a bit frustrating because is not explained clearly how to fill them and does not have a good*

explanation of what each field is about; but it was a good methodology with quite a few tools and I learned different things.

- *Mainly the form that I have reviewed thoroughly has been very easy to fill, but reviewing the others I think they gave me quite a lot of experience in the field of software development and has made me realize the number of documents quite large enough for any project.*
- *The experience was good because we learned how to carry out the necessary aspects for the development of software, as well as all the processes involved.*
- *It was difficult at first because we didn't know well how to fill the forms*
- *It is an orderly and complete way to be able to carry out a project like this, without skipping any step and doing it chronologically.*
- *When making the requirements I could realize that it is the most important thing to make the system*
- *It was difficult to understand how to fill the forms, but once you understood it wasn't so complicated.*
- *The experience was good because we discovered a way to keep track of the processes involved in any project.*
- *Based on the knowledge we were given, we were able to fill out the forms although it was a bit difficult because they are laborious.*
- *In my experience I had never used a methodology like this, it is very effective when using it in the project.*

Question 19: What suggestions could you give to improve the proposed methodology and its tools?

- *Give us more examples of previous forms to guide us to fill them*
- *Make forms friendlier*
- *Make it a little clearer what is required at each point, as there were some misunderstandings during the process.*
- *Reduce the number of fields to fill in some documents*
- *A clearer explanation regarding the use of the tools, especially the forms*
- *Provide a kind of manual at the beginning of the project with all the future stakeholders and with the general requirements, I consider that this would be the best way to work.*
- *I would suggest an instruction manual using more colloquial language if I did not understand the instructions given at first*
- *One of the difficulties is the sequence between the forms and given instructions, that sequence was not very visible and a little dubious.*

Question 20: 100% of the students considered that the methodology achieved the objective of documenting system requirements, adding value to the analysis and design processes, and ensuring the quality of the system.

After obtaining and analyzing the results of the questionnaire as well as taking into account the characteristics of the case studies, we were able to come to certain conclusions. Let us remember that the student projects had some specific characteristics: a) the duration of projects was limited to 6 months, b) projects were real case studies conducted inside small companies, c) projects consisted in designing a system and developing a prototype, d) students had no experience in system design, e) student teams involved students with different backgrounds, mechanical, electrical, software, f) professors played the role of project managers and students played the role of analysis and design team.

We could conclude that the application of the methodology in the case studies confirms that the methodology helps the analysis and design team during the definition of system requirements and that the quality of the finished product is achieved.

In this way, by ensuring that the needs of the stakeholders are well understood, the system will meet their expectations and the product will be a success. If we remember, the costs for modifying the system -in stages subsequent to the conceptual stage- are very high; when DREAM is applied, these additional costs will decrease or, in the best of cases, will be eliminated. In the same way, when applying DREAM, we will avoid changing the requirements due to incorrect identification, which will assure a shorter time of delivery of the product to the market.

The results obtained in the case studies also show that the tools of the methodology are subject to continuous improvement as to simplify forms, to create a glossary with specialized terms, to deeply train the stakeholders in the methodology and its tools, and make the forms more user-friendly and understandable; one possibility, expressed by the students, DREAM could be supported by information technologies.

Finally, we would like to add that our study was limited to a) the circumstances of the communication process, and b) the previously mentioned characteristics of the case studies.

As explained in Chapter 2, in the communication process, there is a difference between what the sender (stakeholder) thinks and expresses, against the message received and interpreted by the receivers (analysis and design team); therefore, we limit ourselves to stating that the methodology helps, considering the communication filters, to the identification of needs and their translation into system requirements.

We realize that the methodology turned out to be a good guide with the conditions present in the case studies; however, questions remains as to what would have happened if the projects had been developed in more time? What would happen if the analysis and design team were a team of experts? What would be the results if it were the design of a complex system? All these variables are exposed and may be the basis for future experimentations.

7.4 Conclusion

This chapter presented the verification and validation of the proposed methodology.

The verification results show that DREAM as a QMS compliance with ISO 15288 (2015), and includes the lean enablers (Oehmen, 2012); finally, DREAM as a QMS possess nine of ten desired quality characteristics of Requirements Engineering models (Batra & Bhatnagar, 2017).

The validation, conducted through case studies, demonstrated that the methodology that we propose helps to the identification of stakeholder needs, their translation into stakeholder requirements, and finally to system requirements. We could evaluate the first prototype of Pill-Bot, resulting in a system that meets user needs. In addition, the risks of project abortion related to the "*unstable, unclear and incomplete requirements*", the second theme of the "*challenges in managing engineering programs*" (Oehmen, 2012), was diminished, due to the fact that stakeholder needs were identified at the beginning of the project.

The questionnaires helped us to identify some opportunities for improvement. According to the student's comments, the DREAM methodology and its support tools are difficult to be applied, they could be improved by being automated by information technologies; stakeholders need to be trained in the use of the methodology; a glossary of technic terms could help to understand easily the methodology.

Next and last chapter presents the conclusions of this research work, as well as the perspectives.

Conclusions & Perspectives

This chapter synthesizes our research work, from the initial problem to the results obtained through the proposal.

Nowadays, in the race of companies to deliver products to the market, there is a challenge to translate the desired quality characteristics into a product. In order to achieve this goal, literature describes "what" is the process of need identification and the translation of needs into requirements, but they do not say "how". Based on this fact, is it possible to methodologically guide the processes of identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product?

Our research work was conducted under Systems Engineering philosophy. In Chapter 1, the problem was located at the Concept stage of the system life cycle, specifically when identifying the stakeholder needs, their translation into stakeholder requirements, and subsequently into system requirements.

The literature review was covered in Chapters 2 to 5. Chapter 2 reviewed Requirements Engineering theory regarding how to express a good requirement statement, its attributes, and classifications. Chapter 3 reviewed how different authors address the problem of translating stakeholder needs into stakeholder requirements, carrying out several activities and tasks. Chapter 4 collects, describes and compares points of views of certain authors, and how they propose to perform the involved activities related to the translation of stakeholder requirements into system requirements.

After reading and analyzing these contributions, we could integrate all relevant contributions into one well-defined process. Nevertheless, we could not find any clue on how to perform some activities or tasks like how to weight the stakeholder level of importance, or what bi-directional traceability at adjacent levels of the system is. Additionally, we found an opportunity to design a lean process to translate stakeholder needs into system requirements; for that reason, we reviewed in Chapter 5 Lean Thinking philosophy and its main fundamentals: value, waste, and create value while reducing waste.

From the lessons learned from the state-of-the-art analysis and our own derived conclusions, in Chapter 6 we proposed a Quality Assurance Methodology for System Requirements definition. We called this methodology DREAM, for Driven Requirements Analysis Management. It supports the two stages of the process: translation from stakeholder needs to stakeholder requirements, and from stakeholder requirements to system requirements. Chapter 6 included DREAM as a quality management system, a set of processes, work instructions, and forms as a complete package to, prior training, can be implemented into an organization.

Chapter 7 verified and validated DREAM. The verification was done through comparing DREAM against ISO 15288 (2015) for quality assurance purposes of the system. The obtained results confirmed that DREAM complies with the standard ISO 15288 (2015). Due to the nature of our research, the validation was carried out through a) *Case Studies* for the application of DREAM as a *quality management system*; and b) *Questionnaires* to obtain data, evaluate the proposal, and subsequently improve it.

Professors and students validated that DREAM helps to identify stakeholder needs and translating them into system requirements; consequently, our methodology contributes to ensuring the quality of a product, reducing or eliminating future costs for changes in the design, and shortening the time to deliver the product to market.

However, students found that DREAM can sometimes be difficult to apply at some steps; they suggested that it would be particularly useful to support the process by an information technology-based tool to decrease the complexity of the proposed forms. They also were interested in having prior training before applying DREAM.

In conclusion, the research question is answered and the expected results have been finally achieved: DREAM and its tools help the understanding of stakeholder needs and their translation into system requirements.

In the course of our research work, we detected very interesting areas to develop future research for further application in industry. For example: stakeholder management, automated quality requirement verification, or application of innovation to eliminate contradictory requirements.

Stakeholder engagement

Communication, participation and collaboration of stakeholders are crucial when conceptualizing a new system. According to Bourne (2015) "*depending on the type of project, between 50% and 90% of the risks […] are associated with stakeholders*" because people are a "*major source of uncertainty*": we are human beings with feelings, values, beliefs, interests, and emotions (Ramos & Berry, 2005; Ramos, Berry, & Carvalho, 2002) that can be translated as opportunities and/or threats and would need to be considered.

Nevertheless, to succeed in this task is not easy. To manage people is one of the most difficult, for not saying *the most difficult* to do; because unfortunately, it is almost impossible to manage the stakeholders that matter. Thus, communication is the key tool that can aid to effectively engage the stakeholders (Bourne, 2015) *"Effective stakeholder engagement = effective stakeholder communication"*.

According to Bourne (2015), effective communication should be designed to be effective within the stakeholder's culture. Every organization has its own culture, and it is necessary to learn how it works, what is normal to the stakeholders, and how to communicate within their paradigm. "*Each cultural grouping exhibits a preferred style of communication, leadership, values and attitudes to work*" (Bourne, 2016).

But what happens if the stakeholders belong to different organizations with different cultures? Are they able to communicate and collaborate?

Savage et al. (2010) suggest that, when the stakeholders belong to multi-sectors, or several organizations, long-term norms must be created for coexistence and communication. This way, stakeholders like investors, suppliers, legislators, government agencies, environmentalist, retailers, the media, special interest groups, and local, state, and federal governments can co-exist.

Why collaboration is so important?

Because collaboration allow to achieve goals that can-not be achieved in any other way; through collaboration social or macro-environmental problems can be tackled; also, organizations gain adaptive advantage due to they learn how to respond in times of uncertainty, environmental complexity, or turbulence (Savage et al. 2010): "*Collaborative advantage refers to the desired synergistic outcome of collaborative activity*".

Savage et al. (2010) mention that there are two strategies to influence stakeholders: coercion (force the cooperation) or compromise (involves offers or concessions to induce cooperation). However, to impose a decision (coercion) by powerful stakeholders should be the last option, because there may be unpredictable consequences for the success of the organization (Ramos, Berry, & Carvalho, 2002).

To develop commitment to collaborate: a) potential partners need to perceive that they are interdependent; b) believe that some advantage can accrue to them by joining forces; and c) develop a common definition of the problem (Savage et al. 2010).

Even if the collaboration is forced by coercion, or gained by compromise, stakeholders need an efficient support tool to communicate.

Stakeholder communication can be supported by some tools as the proposed by Nyamsurena, Lee, Hwang, & Kim (2015), where it is provided a web-based collaborative framework for facilitating decision-making on a 3D design developing process. Their proposal includes the following functions: notification on the surface of 3D data, revision control between the 3D design files, real-time collaboration through the 2D image document, text chatting, and videoconferencing (see Figure 92).

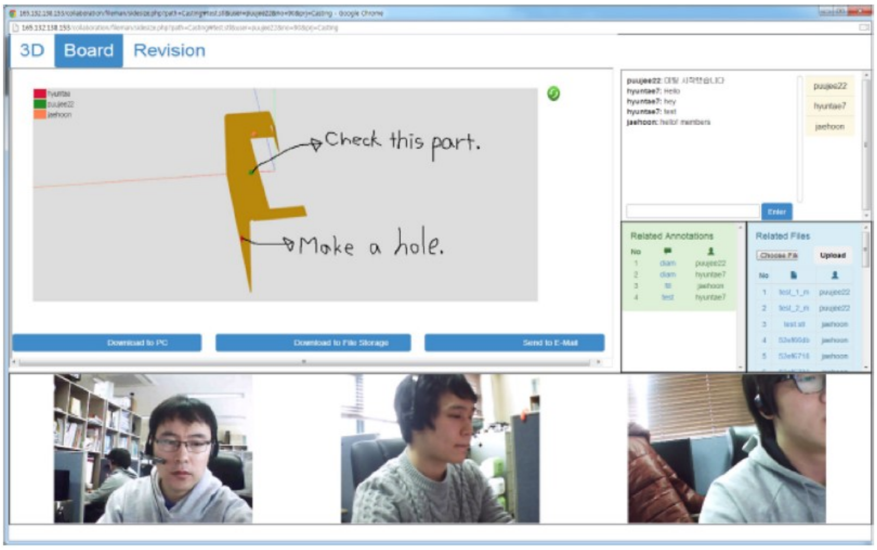


Figure 92 Window of web-based collaborative framework (Nyamsurena, Lee, Hwang, & Kim, 2015)

This way, if we already proposed a methodology to help the understanding of needs and their translation into system requirement, it is quite valuable to work on discovering how to engage the stakeholders, or better said, how to improve and promote their communication, participation, and cooperation. This new effort could lead to the reduction of project risks through stakeholder engagement, and consequently, to the improvement of the conceptualization of a new system and its future acceptance.

Automated quality requirement verification

A key point in communicating requirements is the language. Generally, stakeholders use natural language to express their needs; in spite of there are suggestions of how to express requirements as statements (Guide for writing requirements, 2017) natural language can lead to ambiguities. Unambiguous is one of the quality characteristics that should be present in the requirement statement.

According to Yang et al. (2011) there are four main categories of ambiguity in requirement documents: ambiguity at the level of words (lexical ambiguity), syntax (syntactic ambiguity), semantic interpretation (semantic ambiguity), or the interaction between interpretation and context (pragmatic ambiguity). Nevertheless, Yang et al. (2011) conduct their study to identify automatically sentences which contain anaphoric ambiguity, it is, the risk of misinterpretation between stakeholders.

Currently, a few tools are on the market as the tool "Stimulus for Requirements" "*provides language template libraries […] for writing non ambiguous and standardized system requirements*" (Argosim, 2019); additionally, this tool is very valuable to detect incorrect, contradictory/conflicting, or missing requirements. Nevertheless, there is an opportunity to make some interesting proposals considering that there are several quality characteristics that should be present in a requirement statement, and that requirements should be SMART (Specific, Measurable, Achievable, Realistic, Testable; Faisandier, 2012). To automate quality verification in requirement statements would save time, efforts, and re-work.

Application of innovation to eliminate contradictory requirements.

Once system requirements are defined, the identification of contradictions among them is crucial in order to avoid contradictory design; in case that some requirements are opposites, this contradiction should be resolved before the architecture of the prototype.

But, how to solve contradiction?

Several authors like Altshuller (1984, 2000, 2004), Kraev (2006), Housing (2011), Bukhman (2012), Starovoytova (2015), Russo (2015), and Najari et al. (2016) among others, have worked applying creativity to innovate and resolve problems of contradictory design just in the mid-time of passing through the technical aspect to the physical aspect of a system.

These authors explains their theories and findings applied in different domains; nevertheless, there is a long way to discover how to take advantage of innovation. Sometimes, contradiction is seen as a weakness, it is time to look at it as an opportunity to open our mind to creativity.

We can realize that after applying DREAM, the next stage in the system life cycle is to develop the architecture of the prototype; it is, from the set of requirements resulting from DREAM, different concepts should be explored in parallel, and then should be selected the best possible solution/concept for the system that meets the stakeholder requirements (SEBoK, 2016); at this time, creativity is the door to innovation. Let us take advantage of it.

Until now, we have presented some perspectives for future work; nevertheless, if we look carefully, every task and activity presented in our proposed methodology, we can discover that there are an enormous quantity of opportunities to conduct research; since de negotiation model, passing through the processes of prioritizing requirements, up to the incredible support tools that can be developed by engineers.

Let us imagine, create, and innovate to construct a better world.

Publications

Gomez Sotelo, K. I., Babiker, A., Baron, C., Esteban, P., Gutierrez Estrada, C. Y. A., and Sahraoui, A.-E.-K. (2019). Meta-model for new system development: a case study. 5th IEEE International Symposium on Systems Engineering ISSE 2019, Edinburgh, Scotland, UK, October 1-3. Status: proposed.

Gomez Sotelo, K. I., Baron, C., Esteban, P., Gutierrez Estrada, C. Y. A., & Laredo Velázquez, L. de J. (2018, June). Avez-vous identifie toutes les parties prenantes? *12e Conférence Internationale de Modélisation, Optimisation et Simulation. MOSIM 2018.* Toulouse, France. https://hal.archives-ouvertes.fr/hal-01989427/file/mosim2018.pdf 117-124.

Gómez Sotelo, K. I., Baron, C., Esteban, P., Gutierrez Estrada, C. Y. A., & Laredo Velázquez, L. de J. (2018). How to find non-functional requirements in system developments. *IFAC-PapersOnLine*, *51*(11), 1573–1578. https://doi.org/10.1016/j.ifacol.2018.08.272

Gomez Sotelo, K. I., Baron, C., Esteban, P., & Gutierrez Estrada, C. Y. A. (2017). Metodología para transformar necesidades en requisitos, integrando Ingeniería de Sistemas, Calidad y Pensamiento Esbelto. *Revista de Ingenieria Industrial*, *1*(2), 42–51.

Bibliography

Aasheim, S., & Zhao, Y.-Y. (2017). Developing the stakeholder requirements definition process — A journey of customization. *2017 12th System of Systems Engineering Conference (SoSE)*, 1–6. https://doi.org/10.1109/SYSOSE.2017.7994930

Acosta Flores, J. (2002). *Ingeniería de Sistemas: un enfoque interdisciplinario*. CEIICH-UNAM/Alfaomega.

Adina-Petruța, P., & Roxana, S. (2014). Integrating Six Sigma with Quality Management Systems for the Development and Continuous Improvement of Higher Education Institutions. *Procedia - Social and Behavioral Sciences*, *143*, 643–648. https://doi.org/10.1016/j.sbspro.2014.07.456

AFIS. (2017). AFIS Association Française d'Ingénierie Système. Retrieved from https://www.afis.fr

Akbayrak, B. (2000). A comparison of two data collecting methods: interviews and questionnaires. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, *18*, 1–10.

Alshenqeeti, H. (2014). Interviewing as a Data Collection Method: A Critical Review. *English Linguistics Research*, *3*(1). https://doi.org/10.5430/elr.v3n1p39

Altshuller, G. (2000). *The innovation algorithm: TRIZ, systematic innovation and technical creativity* (1. ed., 2. print). Worcester, Mass: Technical Innovation Center.

Altshuller, G. S. (1984). *Creativity as an exact science: the theory of the solution of inventive problems*. New York: Gordon and Breach Science Publishers.

Altshuller, G., & Shulyak, L. (2004). *And suddenly the inventor appeared: TRIZ, the theory of inventive problem solving* (6. ed). Worcester, Mass: Technical Innovation Center.

Argosim. (2019). Argosim. Retrieved from https://www.argosim.com/home/stimulus-for-requirements/

Atkinson, F. (1990). *Creating Culture Change: The Key to Successful Total Quality Management.* UK: IFS Publications, Milton Keynes.

Bacoup, P., Michel, C., Habchi, G., & Pralus, M. (2015, March). *Implementation of ISO and LEAN management into administration Application in a University Department*. Presented at the 11ème Congrès International Pluridisciplinaire en Qualité, Sûreté de Fonctionnement et Développement Durable – QUALITA'2015, Nancy, France. Retrieved from https://hal.archives-ouvertes.fr/hal-01149800/document

Bacoup, P., Michel, C., Habchi, G., & Pralus, M. (2018). From a Quality Management System (QMS) to a Lean Quality Management System (LQMS). *The TQM Journal*, *30*(1), 20–42. https://doi.org/10.1108/TQM-06-2016-0053

Badreau, S., & Boulanger, J.-L. (2014). *Ingénierie des exigences: Méthodes et bonnes pratiques pour construire et maintenir un référentiel*.

Baines, T., Lightfoot, H., Williams, G. M., & Greenough, R. (2006). State-of-the-art in lean design engineering: A literature review on white collar lean. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *220*(9), 1539–1547. https://doi.org/10.1243/09544054JEM613

Batra, M., & Bhatnagar, D. A. (2017). A Comparative Study of Requirements Engineering Process Model. *International Journal of Advanced Research in Computer Science*, 6.

Bauch, C. (2004). *Lean Product Development: Making waste transparent*. Massachusetts Institute of Technology (MIT) and Technical University of Munich (TUM), Munich.

Berry, D. (2004). *The Requirements Iceberg and Various Icepicks Chipping at It*. Presented at the University of Waterloo, Canada. Retrieved from https://cs.uwaterloo.ca/~dberry/

Berx, K., Friedl, M., Witters, M., & Hehenberger, P. (2016). A customer requirement driven framework for design synthesis - applied to a washing machine. *IFAC-PapersOnLine*, *49*(21), 431–438. https://doi.org/10.1016/j.ifacol.2016.10.642

Blanchard, B. S., & Fabrycky, W. J. (2006). *Systems engineering and analysis* (4th ed). Upper Saddle River, N.J: Pearson Prentice Hall.

Blecken, A., Zobel, A., & Maurantzas, E. (2010). Development of a Lean Quality Management System: An Integrated Management System. In W. Dangelmaier, A. Blecken, R. Delius, & S. Klöpfer (Eds.), *Advanced Manufacturing and Sustainable Logistics* (Vol. 46, pp. 141–151). https://doi.org/10.1007/978-3-642-12494-5_13

Boehm, B. W. (1979). Guidelines for verifying and validating software requirements and design specifications. *EURO IFIP 79*. Presented at the Redondo Beach, CA, USA. https://doi.org/doi:10.1.1.365.8494

Bouchereau, V., & Rowlands, H. (2000). Methods and techniques to help quality function deployment (QFD). *Benchmarking: An International Journal*, *7*(1), 8–19.

Bourne, L. (2015). The three types of stakeholder communication. *PM World Journal*, *IV*(IX), 1–5.

Bourne, L. (2016). Targeted Communication: The Key to Effective Stakeholder Engagement. *Procedia - Social and Behavioral Sciences*, *226*, 431–438. https://doi.org/10.1016/j.sbspro.2016.06.208

Brahm, C., & Kleiner, B. H. (1996). Advantages and disadvantages of group decision- making approaches. *Team Performance Management: An International Journal*, *2*(1), 30–35. https://doi.org/10.1108/13527599610105538

Bukhman, I. (2012). *TRIZ technology for innovation*. Taipei: Cubic Creativity Company.

Bylund, N., Wolf, M., & Mazur, G. (2009, August 24). *Reducing lead time in cutting tool development by implementing Blitz QFD*. *6*, 255–266. Stanford University, Stanford, CA, USA.

Cheng, B. H. C., & Atlee, J. M. (2007). Research Directions in Requirements Engineering. *Future of Software Engineering (FOSE '07)*, 285–303. https://doi.org/10.1109/FOSE.2007.17

Chiarini, A. (2011). Integrating lean thinking into ISO 9001: a first guideline. *International Journal of Lean Six Sigma*, *2*(2), 96–117. https://doi.org/10.1108/20401461111135000

Costello, K., & Ferguson, S. (2010). *Guidelines for Risk Management. S3001. Independent Verification & Validation Program. NASA*.

Cuiller, H. (2015). *System requirements management*. Presented at the INSA Toulouse. INSA Toulouse.

Currie, R. R., Seaton, S., & Wesley, F. (2009). Determining stakeholders for feasibility analysis. *Annals of Tourism Research*, *36*(1), 41–63. https://doi.org/10.1016/j.annals.2008.10.002

Danesh, D., Ryan, M., & Abbasi, A. (2015). Using Analytic Hierarchy Process as a Decision-Making Tool in Project Portfolio Management. *World Academy of Science, Engineering and Technology, International Science Index 108, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, *9*(12), 4194–4204.

Darwish, M., Haque, B., Shehab, E., & Al-Ashaab, A. (2010, September). *Value Stream Mapping and Analysis of product development (engineering) processes*. Presented at the 8th International Conference on Manufacturing Research ICMR 2010, Durham, United Kingdom.

De Weck, O., Eckert, C., & Clarkson, J. (2007, August 28). *A classification of uncertainy for early product and system design*. 12. Paris.

De Weck, O., Rhodes, D., & Ross, A. (2012). Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities). *Third International Engineering Systems Symposium*. Retrieved from https://dspace.mit.edu/handle/1721.1/102927

De Weck, O., & Wilcox, K. (2010). *Multidisciplinary system Design Optimization (MSDO)*. Presented at the Massachussets Institute of Technology. Retrieved from https://ocw.mit.edu/courses/institute-for-data-systems-and-society/ids-338j-multidisciplinary-system-design-optimization-spring-2010/

Faisandier, A. (2012). *Systems Opportunities and Requirements*. France: Sinergy'Com.

Faisandier, A. (2014). *Notions de système et d'ingénierie de système*. France: Sinergy'Com.

Femmer, H. (2017). *Requirements Engineering Artifact Quality: Definition and Control* (Doktors der Naturwissenschaften). Institut für Informatik der Technischen Universität München.

Forsberg, K., & Mooz, H. (1991, October 21). *The Relationship pf System Engineering to the Project Cycle*. Presented at the Join Conference sponsored by National Council On Systems Engineering (NCOSE) and American Society for Engineering Management (ASEM), Chattanooga, TN. Retrieved from http://www.damiantgordon.com/Videos/ProgrammingAndAlgorithms/Papers/The%20Relationship%20of%20System%20Engineering%20to%20the%20Project%20Cycle.pdf

Freeman, R. E., & Reed, D. L. (1983). Stockholders and Stakeholders: A New Perspective on Corporate Governance. *California Management Review*, *25*(3), 88–106. https://doi.org/10.2307/41165018

Glinz, M. (2005). Rethinking the Notion of Non-Functional Requirements. *Proceedings of the Third World Congress for Software Quality*, 55–64.

Glinz, M., & Wieringa, R. (2007). Guest Editors' Introduction: Stakeholders in Requirements Engineering. *IEEE Software*, *24*(2), 18–20. https://doi.org/10.1109/MS.2007.42

Gómez Saavedra, E. (1991). *El Control Total de la Calidad*. Bogotá, Colombia: Fondo Editorial LEGIS.

Grispos, G., García-Galán, J., Pasquale, L., & Nuseibeh, B. (2017). Are you Ready? Towards the Engineering of Forensic-Ready Systems. *11th IEEE International Conference on Research*

Challenges in Information Science, Brighton, United Kingdom. Retrieved from https://arxiv.org/abs/1705.03250

Guide for writing requirements. (2017, June 30). Requirements Working Group International Council on Systems Engineering (INCOSE).

Gutiérrez Pulido, H. (2010). *Calidad total y productividad*. Retrieved from http://site.ebrary.com/id/10820483

Haramis, G. E. (1992). *Implementing a feasibility study "A procedural approach." 16*, 133–138. Great Britain: Pergamon Press Ltd.

Hauser, J. R., & Clausing, D. (1988). The House of Quality. *Harvard Business Review*.

Hickey, A. M., & Davis, A. M. (2003). *Elicitation technique selection: how do experts do it?* 169–178. https://doi.org/10.1109/ICRE.2003.1232748

Hoh In, Boehm, B., Rodger, T., & Deutsch, M. (2001). *Applying WinWin to quality requirements: a case study*. 555–564. https://doi.org/10.1109/ICSE.2001.919130

Hohmann, C. (2019). Value Stream Mapping s'applique-t-elle au développement produit. Retrieved from http://christian.hohmann.free.fr/index.php/lean-en-conception-et-developpement/les-basiques-du-lean-en-conception-et-developpement/652-value-stream-mapping-sapplique-t-elle-au-developpement-produit

Houssin, R., & Coulibaly, A. (2011). An approach to solve contradiction problems for the safety integration in innovative design process. *Computers in Industry*, *62*(4), 398–406. https://doi.org/10.1016/j.compind.2010.12.009

Hughes, R. (2016). *Agile data warehousing for the enterprise: a guide for solution architects and project leaders*. Waltham, MA: Elsevier/MK.

Hull, E., Jackson, K., & Dick, J. (2011). *Requirements engineering* (3rd ed). London ; New York: Springer.

INCOSE. (2017). INCOSE. Retrieved from https://www.incose.org/

ISO 9000 Quality management systems - Fundamentals and vocabulary. (2005). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE).

ISO 9001 Quality management systems - Requirements. (2015). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE).

ISO 15288 Systems and Software Engineering - System Life Cycle Processes. (2015). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE).

ISO 24765 Systems and Software Engineering - System and Software Engineering Vocabulary (SEVocab). (2017). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE).

ISO 29148 Systems and Software Engineering - Life Cycle Processes: Requirements Engineering. (2011). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE).

Jakjoud, A., Zrikem, M., Baron, C., & Ayadi, A. (2013). SysPEM: A SysML and SPEM based process modellinage for systems engineering. *International Journal of Services Operations and Informatics*, *7*(4), 330–348.

Jakjoud, A., Zrikem, M., Baron, C., & Ayadi, A. (2014). SysPEM: Toward a consistent and unified system process engineering metamodel. *Journal of Intelligent Manufacturing*, *27*(1), 149–166. https://doi.org/10.1007/s10845-014-0886-7

Jeong, B. K., & Yoon, T. E. (2016). Improving IT process management through value stream mapping approach: A case study. *Journal of Information Systems and Technology Management*, *13*(3). https://doi.org/10.4301/S1807-17752016000300002

Jepsen, A. L., & Eskerod, P. (2009). Stakeholder analysis in projects: Challenges in using current guidelines in the real world. *International Journal of Project Management*, *27*(4), 335–343. https://doi.org/10.1016/j.ijproman.2008.04.002

Jones, S., Ross, S., & Ruusalepp, R. (2009, May). *Data Audit Framework (DAF) Methodology, draft for discussion, version 1.8*. HATII, University of Glasgow.

Kehl, K., Cain, S., Putze, E., Costello, K., & Kinney, S. (2014). *Risk Review Template. T2006. Independent Verification & Validation Program. NASA*.

Khan, J. A., Rehman, I. ur, Ali, L., Khan, S., Khan, Y. H., & Khan, I. J. (2016). Requirements Prioritization Using Analytic Network Process (ANP). *International Journal of Scientific & Engineering Research*, *7*(11), 35–41.

Kiritani, K., & Ohashi, M. (2015). The Success or Failure of the Requirements Definition and Study of the Causation of the Quantity of Trust Existence Between Stakeholders. *Procedia Computer Science*, *64*, 153–160. https://doi.org/10.1016/j.procs.2015.08.476

Kossiakoff, A., Sweet, W., Seymour, S., & Biemer, S. (2011). *Systems engineering: principles and practice* (2nd ed). Hoboken, N.J: Wiley-Interscience.

Kraev, V. (2006, December). Technical and Physical contradictions - Lesson 3. *The TRIZ Journal*. Retrieved from www.triz-journal.com

Langen, P. H. G. van. (2002). *The anatomy of design: foundations, models and applications*. s.n.], S.l.

Mabrok, M. A., Efatmaneshnik, M., & Ryan, M. J. (2015). Integrating Nonfunctional Requirements Into Axiomatic Design Methodology. *IEEE Systems Journal*, 1–11. https://doi.org/10.1109/JSYST.2015.2462073

Malinin, L. (2016). Design Under Contradictory Requirements. *Volume 11: Systems, Design, and Complexity*, V011T15A025. https://doi.org/10.1115/IMECE2016-65108

Marques, P. A., Meyrelles, P. M., Saraiva, P. M., & Frazao-Guerreiro, F. J. (2016). Integrating Lean Six Sigma with ISO 9001:2015. *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 894–898. https://doi.org/10.1109/IEEM.2016.7798006

Mazur, G. (2002). *QFD Black Belt Notes*. Presented at the United States. United States.

Mazur, G. (2012). *Blitz QFD - The lean aprroach to producto development*. Presented at the ASQ World Conference on Quality and Improvement, Anaheim, CA, USA.

Mazur, G., Terniko, J., & Ziltner, R. (2017). QFD (Quality Function Deployment) Institute. Retrieved from http://www.qfdi.org

Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. *The Academy of Management Review*, *22*(4), 853–886.

Mokammel, F., Coatanéa, E., Coatanéa, J., Nenchev, V., Blanco, E., & Pietola, M. (2018). Automatic requirements extraction, analysis, and graph representation using an approach derived from computational linguistics. *Systems Engineering*, *21*(6), 555–575. https://doi.org/10.1002/sys.21461

Najari, A., Dubois, S., Barth, M., & Sonntag, M. (2016). From Altshuller to Alexander: Towards a Bridge between Architects and Engineers. *Procedia CIRP*, *39*, 119–124. https://doi.org/10.1016/j.procir.2016.01.176

Narendhar, M., & Anuradha, D. K. (2017). Requirement Negotiation Methods in Requirements Engineering. *International Journal of Advanced Research in Computer Science*, 6.

Niebel, B. W., & Freiwalds, A. (2009). *Ingeniería industrial: métodos, estándares y diseño del trabajo*. México, D.F. [etc: McGraw Hill.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering - ICSE '00*, 35–46. https://doi.org/10.1145/336512.336523

Nyamsuren, P., Lee, S.-H., Hwang, H.-T., & Kim, T.-J. (2015). A web-based collaborative framework for facilitating decision making on a 3D design developing process. *Journal of Computational Design and Engineering*, *2*(3), 148–156. https://doi.org/10.1016/j.jcde.2015.02.001

Oehmen, J. (Ed.). (2012). *The Guide to Lean Enablers for Managing Engineering Programs, Version 1.0.* Retrieved from http://hdl.handle.net/1721.1/70495

Palmer, J. D., Liang, Y., & Wang, L. (1990). Classification as an Approach To Requirements Analysis. *Proceedings of the 1st ASIS SIG/CR Classification Research Workshop*, 131–138. https://doi.org/10.7152/acro.v1i1.12472

PMI Project Management Institute. (2017). *Guide to the Project Management Body of Knowledge (PMBOK)*. S.l.: Project Management Institute.

Ramos, I., & Berry, D. M. (2005). Is emotion relevant to requirements engineering? *Requirements Engineering*, *10*(3), 238–242. https://doi.org/10.1007/s00766-005-0014-5

Ramos, I., Berry, D. M., & Carvalho, J. Á. (2002). The Role of Emotion, Values, and Beliefs in the Construction of Innovative Work Realities. In D. Bustard, W. Liu, & R. Sterritt (Eds.), *Soft-Ware 2002: Computing in an Imperfect World* (Vol. 2311, pp. 300–314). https://doi.org/10.1007/3-540-46019-5_22

Rokou, E. (2009). *Development of software tool to support the analytical network process (ANP) for the evaluation of project portfolio* (Thesis). Greek.

Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2008). Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *Systems Engineering*, *11*(3), 246–262. https://doi.org/10.1002/sys.20098

Russo, D., & Duci, S. (2015). From Altshuller's 76 Standard Solutions to a New Set of 111 Standards. *Procedia Engineering*, *131*, 747–756. https://doi.org/10.1016/j.proeng.2015.12.369

Ryan, M. (2013). An Improved Taxonomy for Major Needs and Requirements Artifacts. *INCOSE International Symposium*, *23*(1), 244–258. https://doi.org/10.1002/j.2334-5837.2013.tb03016.x

Ryan, M., & Faulconbridge, I. (2016). *Introduction to Systems Engineering*. Presented at the The University of New South Wales. Retrieved from https://fr.coursera.org/learn/systems-engineering

Ryan, M. J. (2014). Design for system retirement. *Journal of Cleaner Production*, *70*, 203–210. https://doi.org/10.1016/j.jclepro.2014.01.092

Ryan, M. J., Wheatcraft, L. S., Dick, J., & Zinni, R. (2015). On the Definition of Terms in a Requirements Expression. *INCOSE International Symposium*, *25*(1), 169–181. https://doi.org/10.1002/j.2334-5837.2015.00055.x

Ryan, M., & Wheatcraft, L. (2017). On the Use of the Terms Verification and Validation. *INCOSE International Symposium*, *27*(1), 1277–1290. https://doi.org/10.1002/j.2334-5837.2017.00427.x

Saaty, T. L. (2007). Time dependent decision-making; dynamic priorities in the AHP/ANP: Generalizing from points to functions and from real to complex variables. *Mathematical and Computer Modelling*, *46*(7–8), 860–891. https://doi.org/10.1016/j.mcm.2007.03.028

Sadig, A., & Sahraoui, A.-E.-K. (2017). Culture Effect on Requirements Elicitation Practice in Developing Countries. *International Journal of Software Engineering & Applications*, *8*(1), 49–58. https://doi.org/10.5121/ijsea.2017.8105

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2012). *Research methods for business students* (6th ed). Harlow, England ; New York: Pearson.

Savage, G. T., Bunn, M. D., Gray, B., Xiao, Q., Wang, S., Wilson, E. J., & Williams, E. S. (2010). Stakeholder Collaboration: Implications for Stakeholder Theory and Practice. *Journal of Business Ethics*, *96*(S1), 21–26. https://doi.org/10.1007/s10551-011-0939-1

Schillo, R. S., Isabelle, D. A., & Shakiba, A. (2017). Linking advanced biofuels policies with stakeholder interests: A method building on Quality Function Deployment. *Energy Policy*, *100*, 126–137. https://doi.org/10.1016/j.enpol.2016.09.056

SEBoK contributors. (2016). *Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Retrieved from http://sebokwiki.org/w/index.php?title=Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)&oldid=53122

Sharp, H., Finkelstein, A., & Galal, G. (1999). *Stakeholder identification in the requirements engineering process*. 387–391. https://doi.org/10.1109/DEXA.1999.795198

Shukla, V. (2014). *Comprehensive méthodology for the complex systems' requirements engineering & decision making. Computer Aided Engineering* (INSA de Toulouse). Retrieved from https://tel.archives-ouvertes.fr/tel-01168467/document

Sommerville, I., Sawyer, P., & Viller, S. (1998). Viewpoints for requirements elicitation: a practical approach. *Proceedings of ICRE'98*, 74–81.

Starovoytova Madara, D. (2015, July). Theory of inventive problem solving (TRIZ): his-story. *IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 7*. Retrieved from www.ijiset.com

Szejka, A. L., Aubry, A., Panetto, H., Júnior, O. C., & Loures, E. R. (2014). Towards a Conceptual Framework for Requirements Interoperability in Complex Systems Engineering. In R. Meersman, H. Panetto, A. Mishra, R. Valencia-García, A. L. Soares, I. Ciuciu, … H. Chan (Eds.), *On the Move to Meaningful Internet Systems: OTM 2014 Workshops* (Vol. 8842, pp. 229–240). https://doi.org/10.1007/978-3-662-45550-0_24

Tapia Moreno, F. J. (2010). *Cómo elaborar un cuestionario. Nota de Estadística Aplicada a la Administración, Contaduría e Informática Administrativa*. Universidad de Sonora. Departamento de Matemáticas.

Topalović, S. (2015). The Implementation of Total Quality Management in Order to Improve Production Performance and Enhancing the Level of Customer Satisfaction. *Procedia Technology*, *19*, 1016–1022. https://doi.org/10.1016/j.protcy.2015.02.145

Turner, R., & Zolin, R. (2012). Forecasting Success on Large Projects: Developing Reliable Scales to Predict Multiple Perspectives by Multiple Stakeholders Over Multiple Time Frames. *Project Management Journal*, *43*(5), 87–99. https://doi.org/10.1002/pmj.21289

Tyagi, S., Choudhary, A., Cai, X., & Yang, K. (2015). Value stream mapping to reduce the lead-time of a product development process. *International Journal of Production Economics*, *160*, 202–212. https://doi.org/10.1016/j.ijpe.2014.11.002

Van Lamsweerde, A. (2000). Requirements engineering in the year 00: a research perspective. *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, 5–19. https://doi.org/10.1109/ICSE.2000.870392

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., Shortell, T. M., & International Council on Systems Engineering (Eds.). (2015). *Systems engineering handbook: a guide for system life cycle processes and activities* (Fourth edition). Hoboken, New Jersey: Wiley.

Walton, M. (1999). *Strategies for Lean Product Development* (No. WP99-01–91).

Wheatcraft, L. S., & Ryan, M. J. (2018). Communicating Requirements - Effectively! *INCOSE International Symposium*, *28*(1), 716–732. https://doi.org/10.1002/j.2334-5837.2018.00511.x

Wheatcraft, L. S., Ryan, M. J., & Dick, J. (2016). On the Use of Attributes to Manage Requirements. *Systems Engineering*, *19*(5), 448–458. https://doi.org/10.1002/sys.21369

Yamashina, H., Ito, T., & Kawada, H. (2002). Innovative product development process by integrating QFD and TRIZ. *International Journal of Production Research*, *40*(5), 1031–1050. https://doi.org/10.1080/00207540110098490

Yang, H., de Roeck, A., Gervasi, V., Willis, A., & Nuseibeh, B. (2011). Analysing anaphoric ambiguity in natural language requirements. *Requirements Engineering*, *16*(3), 163–189. https://doi.org/10.1007/s00766-011-0119-y

Annexes

Annex 1: ISO 15288 (2015) 6.4.2.3 Activities and tasks

ISO 15288 (2015) 6.4.2.3 Activities and tasks
g) Prepare for Stakeholder Needs and Requirements Definition. This activity consists of the following tasks: *1) Identify the stakeholders who have an interest in the system throughout its life cycle.* *NOTE: This includes individuals and classes of stakeholders who are users, operators, supporters, developers, producers, trainers, maintainers, disposers, acquirer and supplier organizations, parties responsible for external interfacing entities, regulatory bodies and others who have a legitimate interest in the system. Where direct communication is not practicable (e.g., for consumer products and services), representatives or designated proxy stakeholders are selected.* *2) Define the stakeholder needs and requirements definition strategy.* *NOTE: Some stakeholders have interests that oppose the system or oppose each other. When the stakeholder interests oppose each other, but do not oppose the system, this process is intended to gain consensus among the stakeholder classes to establish a common set of acceptable requirements. The intent or desires of those that oppose the system, or detractors of the system, are addressed through the Risk Management process, threat analyses of the System Analysis process, or the system requirements for security, adaptability, or resilience. In this case, the stakeholder needs are not satisfied, but rather addressed in a manner to help ensure system assurance and integrity if actions from the detractors are encountered.* *3) Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition.* *NOTE: This includes the identification of requirements and interfaces for the enabling systems. Enabling systems for stakeholder needs and requirements definition include tools for facilitation and requirements management.* *4) Obtain or acquire access to the enabling systems or services to be used.* *NOTE: The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.*
h) Define stakeholder needs. This activity consists of the following tasks: *1) Define the context of use within the concept of operations and the preliminary life cycle concepts.* *NOTE: Context of use is often captured using a Context of Use Description [ISO/IEC 25063.3]. Preliminary life cycle concepts are developed by the Business or Mission Analysis process.* *2) Identify stakeholder needs.* *NOTE 1: Identification of stakeholder needs includes elicitation of needs directly from the stakeholder, identification of implicit stakeholder needs based on domain knowledge and context understanding, and documented gaps from previous activities. Needs often include measures of effectiveness. Functional analysis is often used to aid the elicitation of needs. Also, quality characteristics of the quality model in ISO/IEC 25010 and quality model application to requirements analysis in ISO/IEC 25030 are useful to elicit and identify quality requirements of non-functional requirements, which are often implicit stakeholder needs.* *NOTE 2 Stakeholder needs describe the needs, wants, desires, expectations and perceived constraints of identified stakeholders. Understanding stakeholder needs for the minimum security and privacy requirements necessary for the operational environment minimizes the potential for disruption in plans, schedules, and performance. If significant issues are likely to arise relating to users and other stakeholders and their involvement in or interaction with a system, recommendations for identifying and treating human-system issues can be found in ISO TS 18152.* *3) Prioritize and down-select needs.* *NOTE: The Decision Management process is typically used to support prioritization. The System Analysis process is used to analyze needs for feasibility or other factors.* *4) Define the stakeholder needs and rationale.* *NOTE: Needs concentrate on system purpose and behavior, and are described in the context of the operational environment and conditions. It is useful to trace needs to their sources and rationale.*
i) Develop the operational concept and other life cycle concepts. This activity consists of the following tasks:

NOTE: Other life cycle concepts can include acquisition concepts, deployment concepts, support concepts, security concepts, and retirement concepts. In this activity, the preliminary life cycle concepts defined within the Business or Mission Analysis process are further developed in the context of specific stakeholder needs, as associated scenarios and interactions are defined. See ISO/IEC/IEEE 29148 clauses 5 and 6 for more information on operational concepts, and ISO/IEC/IEEE 29148 Annex A for an annotated outline for a System Operational Concept.

1) Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other life cycle concepts.

NOTE 1: Scenarios are used to analyze the operation of the system in its intended environment in order to identify additional needs or requirements that possibly have not been explicitly identified by any of the stakeholders, e.g., legal, regulatory and social obligations. The context of use of the system is identified and analyzed, including the activities that users perform to achieve system objectives, the relevant characteristics of the end users of the system (e.g., expected training, degree of fatigue), the physical environment (e.g., available light, temperature) and any equipment to be used (e.g., protective or communication equipment). The social and organizational influences on users that could affect system use or constrain its design are analyzed when applicable. Scenarios centered on attackers, their environments, tools, techniques, and capabilities are key considerations for operational concept development.

Scenarios are prioritized in order to reflect the weighted importance of the various operational needs.

NOTE 2: These scenarios often motivate updates to the operational or other life cycle concepts. Abuse and failure scenarios highlight the need for additional functional requirements (or more specific derived requirements) to mitigate risks that are identified in the abuse or failure scenarios.

2) Identify the interaction between users and the system.

NOTE 1: Usability requirements take into account human capabilities and skills limitations. Where possible, applicable standards, e.g., ISO 9241, and accepted professional practices are used in order to define:

i) Physical, mental, and learned capabilities.

ii) Work place, environment, and facilities, including other equipment in the context of use.

iii) Normal, unusual, and emergency conditions.

iv) Operator and user recruitment, training and culture.

NOTE 2: If usability is important, usability requirements are planned, specified, and implemented through the life cycle processes. Refer to ISO TS 18152 for information on human-system issues and ISO/IEC 25060:2010 for information on usability.

j) Transform stakeholder needs into stakeholder requirements. This activity consists of the following tasks:

1) Identify the constraints on a system solution.

NOTE: These result from 1) instances or areas of stakeholder-defined solution; 2) implementation decisions made at higher levels of system hierarchical structure; 3) required use of defined enabling, legacy, or interfacing systems or system elements, resources and staff; or 4) stakeholder defined affordability objectives. Include those that are unavoidable consequences of existing agreements, management decisions, and technical decisions.

2) Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health.

NOTE 1: See ISO/IEC/IEEE 15026 for additional information on system and software assurance.

NOTE 2: Identifying safety risks facilitates the identification of safety requirements and functions. Safety risks include those associated with methods of operations and support, health and safety, threats to property and environmental influences. Use applicable standards, e.g., IEC 61508, and accepted professional practices.

NOTE 3: Identifying security risks facilitates the identification of additional security requirements and functions. If warranted, include applicable areas of system security, including physical, procedural, communications, computers, programs, data and emissions. This includes access and damage to protected personnel, properties and information, compromise of sensitive information, and denial of approved access to property and information. This also includes the required security functions, such as mitigation and containment, referencing applicable standards and accepted professional practices where mandatory or relevant.

NOTE 4: See ISO/IEC 25030 for further information regarding quality characteristics from a quality in use perspective.

3) Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics.

NOTE 1: See ISO/IEC/IEEE 29148 clauses 5 and 6 for more information on stakeholder requirements, and clauses 8 and 9 for a description of and an annotated outline for a Stakeholder Requirements Specification.

NOTE 2: The stakeholder requirements are reviewed at key decision times in the life cycle to help ensure that account is taken of any changes of need.

NOTE 3: The stakeholder requirements are recorded in a form suitable for requirements management through the life cycle. These records establish the stakeholder requirements baseline, and retain changes of need and their origin throughout the system life cycle. These records are the basis for traceability to decisions made by the Business or Mission Analysis process as well as stakeholder needs, system requirements, and subsequent system elements.

NOTE 4: The stakeholder requirements are the basis of the validation criteria for the system and system elements.

k) Analyze stakeholder requirements. This activity consists of the following tasks:

1) Analyze the complete set of stakeholder requirements.

NOTE 1: Stakeholder requirements are analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded. ISO/IEC/IEEE 29148 provides additional information on characteristics of requirements.

NOTE 2: The System Analysis process is used to assess feasibility and affordability. The Verification and Validation processes are used in the review of stakeholder requirements.

2) Define critical performance measures that enable the assessment of technical achievement.

NOTE: This includes defining technical and quality measures and critical performance parameters associated with each effectiveness measure identified in the stakeholder requirements. The critical performance measures (e.g., measures of effectiveness and measures of suitability) are defined, analyzed and reviewed to help ensure stakeholder requirements are met and to help ensure identification of project cost, schedule or performance risk associated with any non-compliance. ISO/IEC 15939 provides a process to identify, define and use appropriate measures. INCOSE TP-2003-020-01, Technical Measurement, provides information on the selection, definition, and implementation of critical performance measures. The ISO/IEC 25000 series of standards provides relevant quality measures.

3) Feedback the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed.

4) Resolve stakeholder requirements issues.

NOTE: This includes requirements that violate the characteristics for individual requirements or the set of requirements as defined in ISO/IEC/IEEE 29148.

f) Manage the stakeholder needs and requirements definition. This activity consists of the following tasks:

1) Obtain explicit agreement on the stakeholder requirements.

NOTE: This includes confirming that stakeholder requirements are expressed correctly, comprehensible to originators, and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

2) Maintain traceability of stakeholder needs and requirements.

NOTE: Through the life cycle, bi-directional traceability is maintained between the stakeholder needs and requirements and the stakeholders and sources, organizational strategy, and business and mission problems and opportunities. Additional traceability to systems making up the system solution facilitates the transition to the System Requirements Definition process. This is often facilitated by an appropriate data repository.

3) Provide key information items that have been selected for baselines.

NOTE: The Configuration Management process is used to establish and maintain configuration items and baselines. This process (Stakeholder Needs and Requirements Definition) identifies candidates for the baseline, and then provides the information items to CM. For this process, the stakeholder needs, stakeholder requirements, and operational concept are typical information items that are baselined.

Annex 2: ISO 15288 (2015) 6.4.3.3 Activities and tasks

ISO 15288 (2015) Activities and tasks
e) ***Prepare for System Requirements Definition. This activity consists of the following tasks:*** *1) Define the functional boundary of the system in terms of the behavior and properties to be provided.* *NOTE The functional boundary definition is partly based on the context of use and operational scenarios defined in the frame of the Stakeholder Needs and Requirements Definition process. This includes the system's stimuli and its responses to user and environment behavior, and an analysis and description of the required interactions between the system and its environment in terms of interface properties and constraints, such as mechanical, electrical, mass, thermal, data, and procedural flows. This establishes the expected system behavior, expressed in quantitative terms, at its boundary.* *2) Define the system requirements definition strategy.* *NOTE This includes the approach to be used to identify and define the system requirements, and manage the requirements through the life cycle.* *3) Identify and plan for the necessary enabling systems or services needed to support system requirements definition.* *NOTE This includes identification of requirements and interfaces for the enabling systems. Enabling systems for system requirements definition include tools for facilitation and requirements management.* *4) Obtain or acquire access to the enabling systems or services to be used.* *NOTE: The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.*
f) ***Define system requirements. This activity consists of the following tasks:*** *1) Define each function that the system is required to perform.* *NOTE 1 This includes how well the system, including its operators, is required to perform that function, the conditions under which the system is to be capable of performing the function, the conditions under which the system is to commence performing that function and the conditions under which the system is to cease performing that function. In some cases, functions are derived from analysis of critical quality characteristics (e.g., system diagnosing function or highly frequent data backup function for reliability).* *NOTE 2 Conditions for the performance of functions can incorporate reference to required states and modes of operation of the system. System requirements depend heavily on abstract representations of proposed system characteristics and sometimes employ multiple modeling techniques and perspectives to give a sufficiently complete description of the desired system requirements.* *NOTE 3 Enabling functions that are required to support the system-of-interest in achieving its functionality are also identified and defined concurrently with the function of the system-of-interest. This is necessary to help ensure that the enabling functions are identified and accounted for.* *2) Define necessary implementation constraints.* *NOTE This includes the implementation decisions that are allocated from architecture definition at higher levels in the structure of the system and are introduced by stakeholder requirements or are solution limitations.* *3) Identify system requirements that relate to risks, criticality of the system, or critical quality characteristics.* *NOTE Critical quality characteristics commonly include those related to health, safety, security assurance, reliability, availability, and supportability. This includes analysis and definition of safety considerations, including those relating to methods of operation and maintenance, environmental influences and personnel injury. It also includes helping to ensure each safety related function and its associated safety integrity, is expressed in terms of the necessary risk reduction, and is specified and allocated to designated safety-related systems. Applicable standards are used concerning functional safety, e.g., IEC 61508, and environmental protection, e.g., ISO 14001. Analyze security considerations including those related to compromise and protection of sensitive information, data, and material. The security-related risks are defined, including, but not limited to, administrative, personnel, physical, computer, communication, network, emission and environment factors using, as appropriate, applicable security standards. Refer to ISO/IEC/IEEE 15026-4 for system and software assurance guidance. ISO/IEC 27036 provides guidance for information security requirements for the outsourcing of products and services. ISO 25030 provides guidance for external system quality factors and characteristics. For systems intended for human interaction, human factors engineering (ergonomics) specifications are considered. For systems that have usability requirements, recommendations for obtaining a desired level of usability can be found in ISO TR 18529, Ergonomics – Ergonomics of human-system interaction – Human-centered life cycle process descriptions.* *4) Define system requirements and rationale.* *NOTE 1 This includes defining system requirements consistent with stakeholder requirements, functional boundaries, functions, constraints, cost targets, identified interfaces, and critical quality characteristics. Consistent practice has shown this process requires iterative and recursive steps in parallel with other life cycle processes through the system hierarchy. See ISO/IEC/IEEE 29148 clauses 5*

and 6 for more information on system requirements, and clauses 8 and 9 for a description of and an annotated outline for a System Requirements Specification.

NOTE 2 The system requirements are recorded in a form suitable for requirements management through the life cycle. These records establish the system requirements baseline, and include the associated rationale, decisions, and assumptions. They are the basis for traceability to information items and subsequent system elements. Change requests of system requirements also provide a rationale to help in the determination of the acceptability of the proposed change, including consistency with stakeholder requirements.

NOTE 3 The System Analysis Process is used to determine appropriate values for requirement parameters, considering the estimated cost, schedule, and technical performance of the system. The Validation Process is used to determine if the requirements address the stakeholders' needs. The Verification Process determines the quality of the requirements with respect to the attributes and characteristics of good requirements (refer to ISO/IEC/IEEE 29148).

g) Analyze system requirements. This activity consists of the following tasks:

1) Analyze the complete set of system requirements.

NOTE 1 System requirements are analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded. ISO/IEC/IEEE 29148 provides additional information on characteristics of requirements. Deficiencies, conflicts, and weaknesses are identified and resolved within the complete set of system requirements.

NOTE 2 The System Analysis process is used to assess feasibility, affordability, and balance.

2) Define critical performance measures that enable the assessment of technical achievement.

NOTE This includes defining technical and quality measures and critical performance parameters associated with each effectiveness measure identified in the system requirements. The critical performance measures (e.g., measures of performance and technical performance measures) are analyzed and reviewed to help ensure system requirements are met and to help ensure identification of project cost, schedule or performance risk associated with any non-compliance. ISO/IEC 15939 provides a process to identify, define and use appropriate measures. INCOSE TP-2003-020-01, Technical Measurement, provides information on the selection, definition, and implementation of critical performance measures. The ISO/IEC 25000 series of standards provides relevant quality measures.

3) Feedback the analyzed requirements to applicable stakeholders for review.

NOTE Feedback helps to ensure that the specified system requirements have been adequately captured and expressed. Confirmation is made that they are a necessary and sufficient response to stakeholder requirements and a necessary and sufficient input to other processes, in particular architecture and design. This is one application of the Validation Process applied for the specific requirements.

4) Resolve system requirements issues.

NOTE This includes requirements that violate the characteristics for individual requirements or the set of requirements as defined in ISO/IEC/IEEE 29148.

h) Manage system requirements. This activity consists of the following tasks:

NOTE Maintaining system requirements includes defining, recording, and controlling the baseline, generally under formal configuration management, along with managing any changes resulting from the application of other life cycle processes such as architecture or design.

1) Obtain explicit agreement on the system requirements.

NOTE This includes confirming that system requirements are expressed correctly, comprehensible to originators, and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

2) Maintain traceability of the system requirements.

NOTE Through the life cycle, bi-directional traceability is maintained between the system requirements and the stakeholder requirements, architecture elements, interface definitions, analysis results, verification methods or techniques, and allocated, decomposed, and derived requirements. This helps ensure that all achievable stakeholder requirements are met by one or more system requirements, and all system requirements meet or contribute to meeting at least one stakeholder requirement. This is often facilitated by an appropriate data repository.

3) Provide key information items that have been selected for baselines.

NOTE The Configuration Management process is used to establish and maintain configuration items and baselines. This process (System Requirements Definition) identifies candidates for the baseline, and then provides the information items to CM. For this process, the system requirements are typical information items that are baselined.

Annex 3: DREAM as a QMS

As a Quality Management System

Content

Introduction

Currently, enterprises are competing among them trying to win the race of delivering new products to market. In the way, some of them mistake the user needs and provide articles that do not satisfy consumers; frequently, the resulting product does not accomplish the purpose for which it was requested, designed, and manufactured.

The present document was developed as part of the thesis: "Quality Assurance Methodology for System Requirements definition", centered in answer the question: is it possible to methodologically guide the processes of identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product?

In order to answer the question we developed a methodology and its tools, based on Systems Engineering and in compliance with the quality standard ISO 15288 (2015), to ensure, as far as possible, that the stakeholder needs are understood, and later, are translated into system requirements, reducing risks of project abortion or delays, and adding value to the analysis and design process.

The methodology was named DREAM, meaning Driven Requirement Analysis Management; its tools are presented here:

DREAM as a quality management system (QMS)

DREAM as a QMS consists of the following documentation: two processes (P), eleven work instructions (W) and twenty forms (F). The processes describe the flux of activities to be performed, the work instructions show how to perform these activities, and the forms provide support to record all information resulting from conducting the activities.

DREAM as a QMS was validated through some case studies, student projects that possessed some specific characteristics: a) they were completed in one semester, b) some of them were conducted inside a small enterprise, c) they allowed the conceptualization of a system and the development of a first prototype, d) involved students with no experience in system conceptualization, e) the involved students had different domain expertise like mechanical, electrical, or software engineering.

These conditions allowed us: a) to test DREAM in a short period, several times in parallel, and in different domains, b) to validate the prototype with real stakeholders, and c) to identify improvement opportunities for DREAM as a QMS. The results showed that students found DREAM useful as a methodological guide for identifying stakeholder needs and translating them into system requirements to assure, as far as possible, the quality of a product.

The reader is invited to know DREAM as a QMS, follow the processes and apply these tools that will provide help when identifying stakeholder needs and translate them into system requirements.

DREAM Processes

P–1 Stakeholder Needs to Stakeholder Requirements

DREAM	**Process** **Stakeholder Needs to Stakeholder Requirements**	ID Process P – 1 Version 001

1. Objective
To identify and weight the stakeholders
To define system mission, purpose, and objectives
To elicit stakeholder needs (SkN)
To verify stakeholder needs feasibility and consistency
To prioritize stakeholder needs
To translate stakeholder needs into stakeholder requirements (SkR)
To list stakeholder requirements
To create a single record for every stakeholder requirement

2. Responsible
The Project Manager is responsible for the correct application of this Process (P)

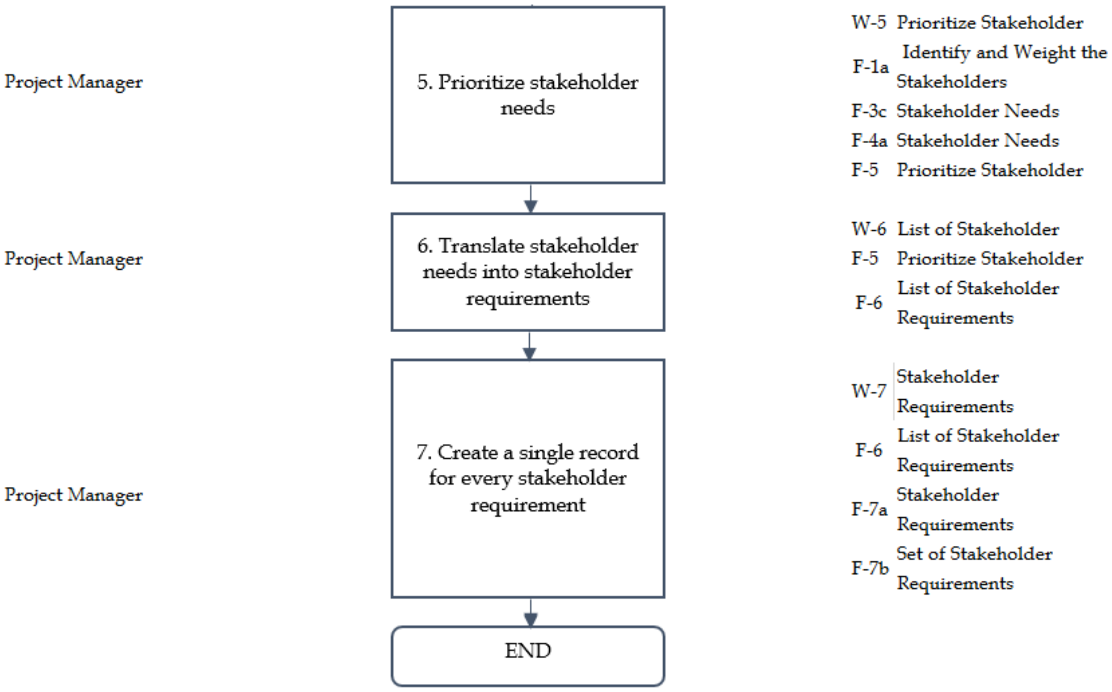
3. Scope
This P may be applied to New System Development projects

4. Definitions
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities
Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram

Responsible	Activity	Documents
	START	
Project Manager	1. Identify and weight the stakeholders	W-1 Identify and Weight the Stakeholders F-1a Identify and Weight the Stakeholders F-1b List of Stakeholders
Project Manager	2. Define system mission, purpose and objectives	W-2 Define system mission, purpose and objectives F-2a Define system mission, purpose and objectives F-2b Collaborative session to define system mission, purpose and objectives
Project Manager	3. Elicit stakeholder needs	W-3 Elicit Stakeholder Needs F-3a Elicit Stakeholder Needs F-3b Collaborative Session Elicit Stakeholder Needs F-3c Stakeholder Needs
Project Manager	4. Verify stakeholder needs feasibility and consistency	W-4 Stakeholder Needs F-1a Identify and Weight the Stakeholders F-3c Stakeholder Needs F-4a Stakeholder Needs F-4b Stakeholder Negotiation



7. Documents

W - 1	Identify and Weight the Stakeholders
F – 1a	Identify and Weight the Stakeholders
F – 1b	List of Stakeholders
W - 2	Define system mission, purpose, and objectives
F – 2a	Define system mission, purpose, and objectives
F – 2b	Collaborative session to define system mission, purpose, and objectives
F – 2c	System mission, purpose and objectives
W - 3	Elicit Stakeholder Needs
F – 3a	Elicit Stakeholder Needs
F – 3b	Collaborative Session Elicit Stakeholder Needs
F – 3c	Stakeholder Needs
W - 4	Stakeholder Needs feasibility and consistency
F – 4a	Stakeholder Needs feasibility and consistency
F – 4b	Stakeholder Negotiation
W - 5	Prioritize Stakeholder Needs
F - 5	Prioritize Stakeholder Needs
W - 6	List of Stakeholder Requirements
F - 6	List of Stakeholder Requirements
W - 7	Stakeholder Requirements
F – 7a	Stakeholder Requirements
F – 7b	Set of Stakeholder Requirements

8. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

P–2 Stakeholder Requirements to System Requirements

DREAM	Process **Stakeholder Requirements to System Requirements**	ID Process P – 2 Version 001

1. Objective

To analyze stakeholder requirements (SkR)
To translate stakeholder requirements into system requirements (SyR)
To list system requirements
To identify bi-directional traceability at adjacent levels of the system hierarchy
To create a single record for every system requirement

2. Responsible

The Project Manager is responsible for the correct application of this Process (P)

3. Scope

This P may be applied to New System Development projects

4. Definitions

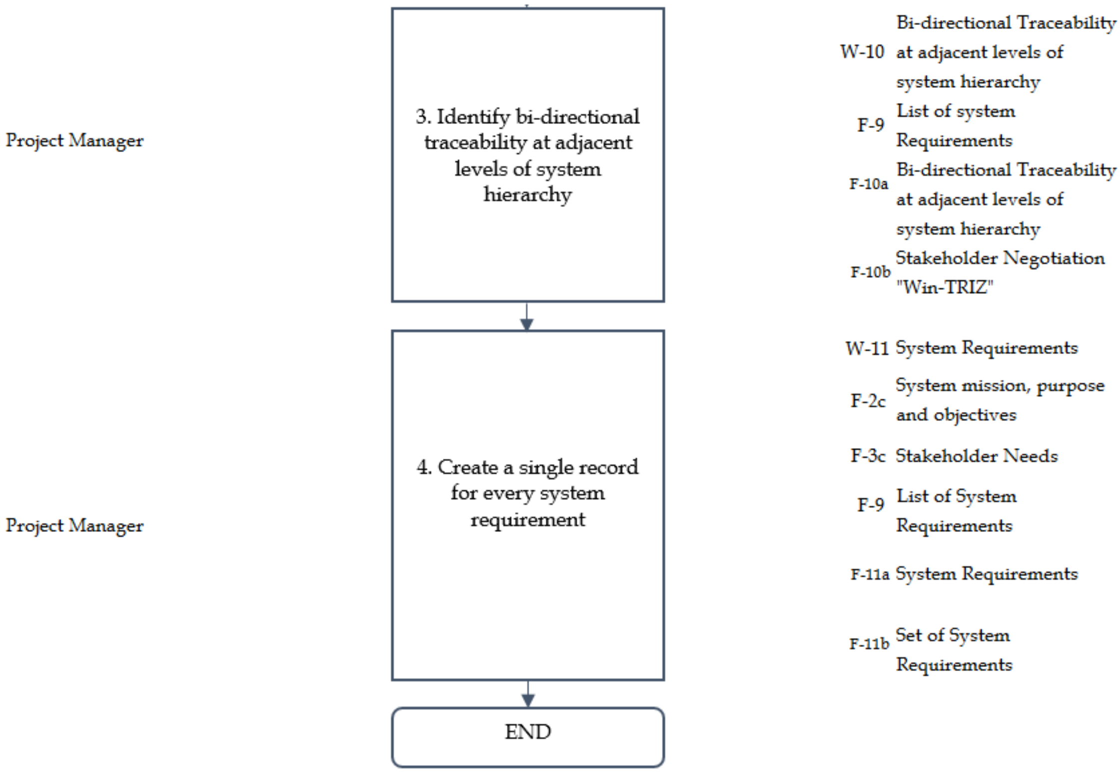
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram

Responsible	Activity	Documents
	START	
Facilitator	1. Analyze stakeholder requirements	W-8 Analyze Stakeholder Requirements F-1a Identify and Weight the Stakeholders F-1b List of Stakeholders F-2c System mission, purpose and objectives F-3c Stakeholder Needs F-4a Stakeholder Needs feasibility and consistency F-4b Stakeholder Negotiation F-5 Prioritize Stakeholder Needs F-6 List of Stakeholder Requirements F-7b Set of Stakeholder Requirements F-8 Analyze Stakeholder Requirements
Project Manager	2. Translate stakeholder requirements into system requirements	W-9 List of System Requirements F-6 List of Stakeholder Requirements F-9 List of System Requirements



7. Documents

F – 1a	Identify and Weight the Stakeholders
F – 1b	List of Stakeholders
F – 2c	System mission, purpose and objectives
F – 3c	Stakeholder Needs
F – 4a	Stakeholder Needs feasibility and consistency
F – 4b	Stakeholder Negotiation
F - 5	Prioritize Stakeholder Needs
F - 6	List of Stakeholder Requirements
F – 7b	Set of Stakeholder Requirements
W - 8	Analyze Stakeholder Requirements
F – 8	Analyze Stakeholder Requirements
W - 9	List of System Requirements
F – 9	List of System Requirements
W - 10	Bi-directional Traceability at adjacent levels of the system hierarchy
F – 10a	Bi-directional Traceability at adjacent levels of the system hierarchy
F – 10b	Stakeholder Negotiation "Win-TRIZ"
W - 11	System Requirements
F – 11a	System Requirements
F – 11b	Set of System Requirements

8. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

DREAM Work Instructions

W-1 Identify and Weight the Stakeholders

DREAM	**Work Instruction** **Identify and Weight the stakeholders**	ID Work Instruction W – 1 Version 001

1. Objective

To identify the stakeholders across the life cycle
To weight the stakeholders according to their power, legitimacy, and urgency

2. Responsible

The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope

This W may be applied to New System Development projects

4. Definitions

System Life Cycle stages and their purposes:
Concept: Identify Stakeholders needs; evaluate alternate concepts; recommend possible solutions
Development: Develop detail planning; Identify and manage risk and business opportunities
Production: Produce systems; inspect and test
Utilization: Operate system to satisfy user's needs
Support: Provide sustained system capability
Retirement: Store, archive or dispose of system

Stakeholder Types:
Legislators: Professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for operation that will affect the development and/or operation of the system.
Owner or investor: Who pay, by, pay for its operation, and obtain revenue
Project executive or project sponsor: Prior to the Project, identify the need for a new asset and the potential benefit it will bring
Consumers: Who buy the product the new assets produces
Operators/users: Who operate the asset on behalf of the owner
Project manager and Project team: Who manage the project, the analysis and design team in the requirement engineering process
Senior supplier (design and/or management): Senior management in the lead contractor
Other suppliers: People or groups who provide goods, materials; works or services
Public: The public concerned with environmental and social impacts of the system, wanted to know how their taxes have been spent
People who don't want the system or maliciously intended (to identify possible risks or threats)

Stakeholder attributes:
Power: A relationship among social actors in which one social actor, A, can get another social actor, B, to do something that B would not have otherwise done
Legitimacy: A generalized perception or assumptions that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values, beliefs, definitions
Urgency: The degree to which stakeholder claims call for immediate attention

Facilitator: Who leads the collaborative work session
Participants: The analysis and design team
Stakeholder role: The role who plays the Stakeholder; example: Project Manager, Operator, Inspector, Auditor, etc.

Table 1. Rubric to weight the Stakeholders

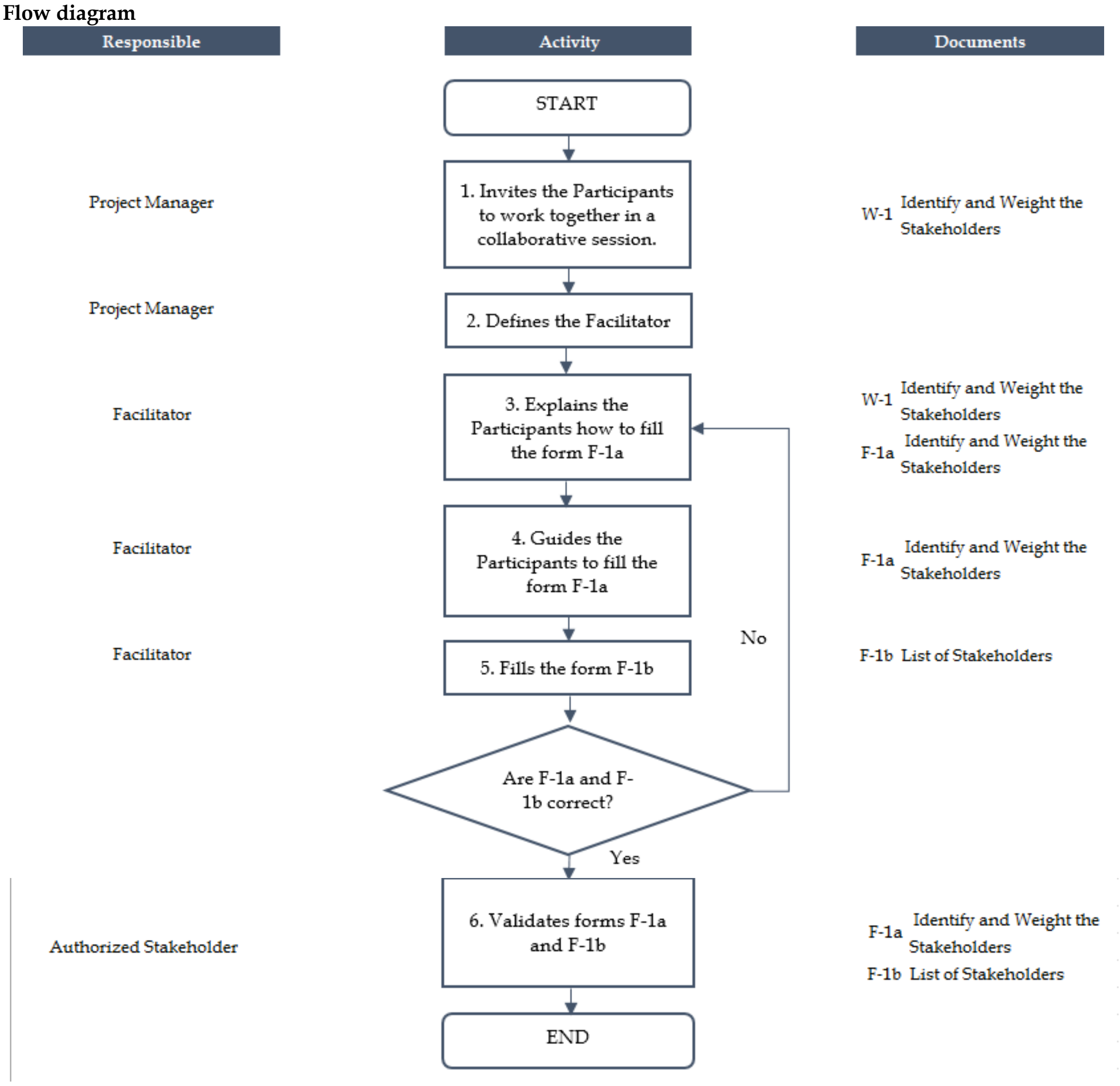
Proposed value	0	1	2	3
Power	This stakeholder cannot influence or make decisions to determine the actions to follow	Sometimes the stakeholder may influence the decisions to determine the actions to follow	Sometimes the stakeholder can make decisions and/or determine the actions to follow	The power is absolute, this stakeholder can make decisions and/or determine the actions to follow
Proposed value	**0**	**0,5**	**1**	**1,5**
Legitimacy				

	Stakeholders that don't have a legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have a little legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have some legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome	Stakeholders that have total legal, moral, or pre-assumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome
Proposed value	0	0,25	0,5	0,75
Urgency	Managerial delay in attending to the claim or relationship is acceptable to the stakeholder AND/OR Stakeholders who don't claim for immediate attention	Managerial delay in attending to the claim or relationship is sometimes acceptable to the stakeholder AND/OR Stakeholders who few times claim for immediate attention	Managerial delay in attending to the claim or relationship is almost never acceptable to the stakeholder AND/OR Stakeholders who almost ever claim for immediate attention	Managerial delay in attending to the claim or relationship is unacceptable to the stakeholder AND/OR Stakeholders who claim for immediate attention

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram



7. Description

1. The Project Manager reunites the participants to work together in a collaborative session.
2. The Project Manager defines who is going to be the Facilitator
3. The Facilitator explains everyone the fields of the form *F-1a Identify and value the Stakeholders.* This form contains the following fields: a) at the top, the different stages of the system life cycle; in each stage different columns, stakeholders, power, legitimacy and urgency values, and a column for the total stakeholder value; b) at the left side, the different types of stakeholders.
4. The Facilitator guides the Participants to fill the form *F-1a Identify and weight the Stakeholders*

Instructions to fill the form F-1a Identify and weight the Stakeholders:

4.1 All the Participants, guided by the Facilitator identify the stakeholders according to the stakeholder typology and write them down in the column stakeholders of the form. All the Participants, guided by the Facilitator identify the stakeholders according to the stakeholder typology and write them down in the column stakeholders of the form.
4.2 The Facilitator chose one identified stakeholder and read each one of the four descriptions of the power attribute shown in Table 1.
4.3 Based on knowledge and experience, the Participants select the description that better fits the stakeholder. Only one description should be chosen, so, the team discusses their propositions until a consensus is found.
4.4 The Facilitator writes in the form the corresponding value of the chosen description.
4.5 Repeat steps 4.2 to 4.4 with legitimacy and urgency attributes
4.6 The Facilitator obtains the stakeholder total weight by adding the values of power, legitimacy, and urgency, write it down in the column T.
4.7 Repeat steps 4.2 to 4.6 until weight all stakeholders

The total stakeholder weight is obtained through the addition of the values obtained in each attribute. For example: if the stakeholder “x”:

a) …sometimes can make decisions and/or determine the actions to follow; then, according to the reference table, his power value is 2.00
b) … have a total legal, moral, or presumed claim on the firm and groups that have an ability to influence the firm's behavior, direction, process, or outcome; then, according to the reference table, his legitimacy value is 1.50
c) … don’t claim for immediate attention, then; according to the reference table, his urgency value is 0.00

This way, the total stakeholder “x” weight is equal to 2.00 + 1.50 + 0.00 = 3.50
With the proposed values, the maximum possible total stakeholder weight is 5.25 obtained as following: maximum power value 3.00, plus maximum legitimacy value 1.50, plus maximum urgency value 0.75.

NOTE 1: When the total stakeholder weight is equal to zero, this person or organization is a non-stakeholder, she/he/it has to be removed from the stakeholder group.
NOTE 2: It is possible that a stakeholder appears several times in the system life cycle, and it is possible that the weight obtained in each stage are different; to solve this possibility, it is suggested to take into account the Maximum Total Stakeholder Value, shown in the last column of F-1a Identify and value the stakeholders.

5. The Facilitator fills the form F-1b List of Stakeholders
6. The Facilitator and the Participants verify that forms *F-1a Identify and value the Stakeholders* and *F-1b List of Stakeholders* are correct and the authorized stakeholder validate them.

8. Documents

F - 1a Identify and Weight the Stakeholders
F - 1b List of Stakeholders

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-2 Define system mission, purpose, and objectives

DREAM	**Work Instruction** **Define system mission, purpose, and objectives**	ID Work Instruction W – 2 Version 001

1. Objective

To define the problem in terms that all the Participants understand it
To define the system mission, purpose, and objectives
To analyze the system contextual situation and the different operational and incident system scenarios
To define the system operational and technical concept
To state where the value is for the customers and what is considered success for the project
To consider the underlying assumptions upon which the system are based, the possible opportunities for improvements to be delivered by the system, and the implementation details relating to the system
To identify risks and uncertainty that could affect the future success of the system

2. Responsible

The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope

This W may be applied to New System Development projects

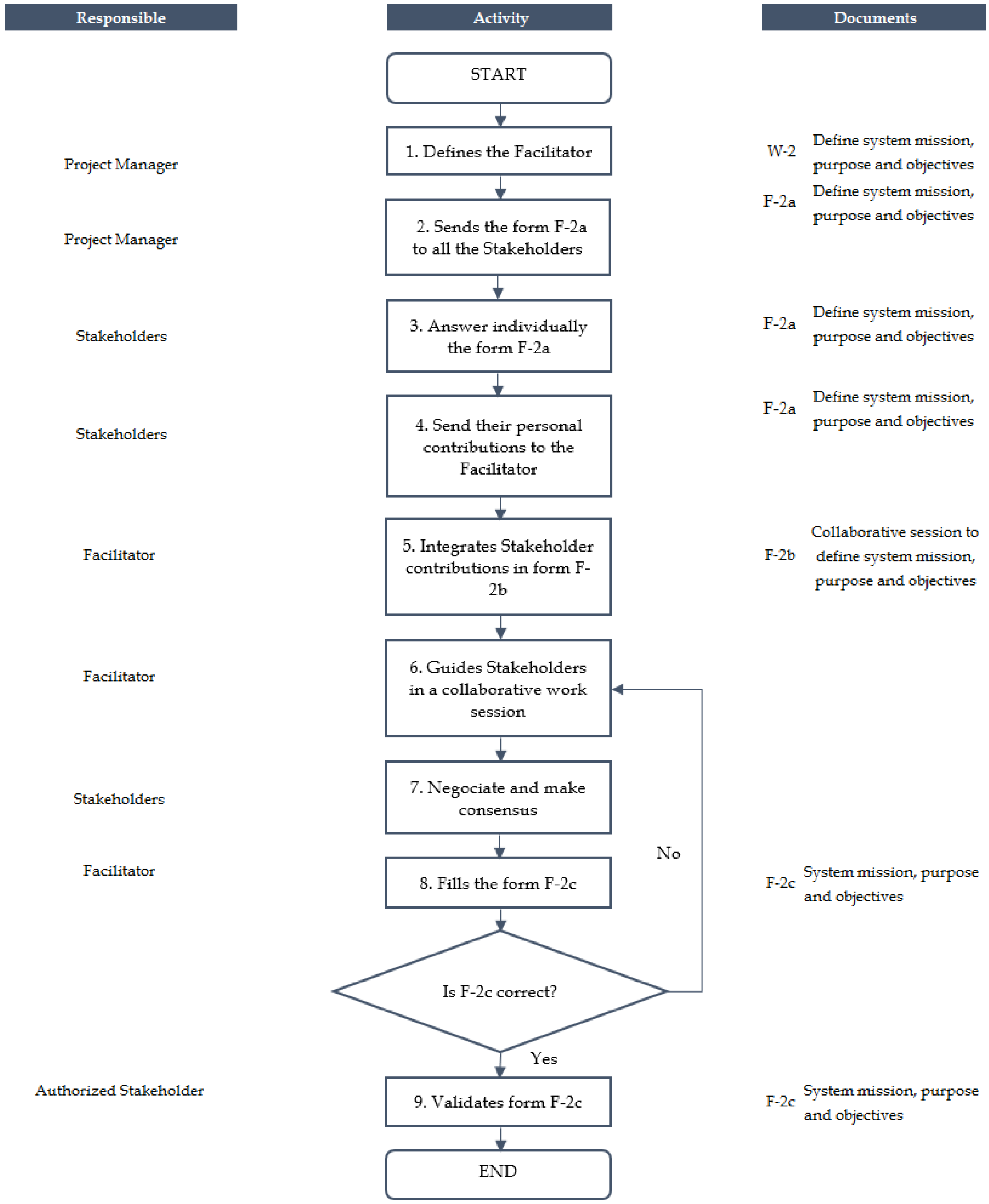
4. Definitions

System of Interest (SoI): The system whose life cycle is under consideration.
Context: Describes the system relationships and environment, resolved around a selected system-of-interest.
Scenario: A set of actions/functions representing the dynamic of exchanges between the functions allowing the system to achieve a mission or a service.
State or Mode: A description of the current or future potential, or the health of the system.
System mission: The top-level function of the system; the one that synthesizes all transformation of all inputs and solicitations into outputs and reactions.
System purpose: What the system is for, and why the different stakeholders are willing to participate in the system lifecycle.
Operational scenarios: Description of an imagined sequence of events that includes the interaction of the product or service with its environment and users, as well as interaction among its product or service components.
Value: Is what the customer considers important, and its willing to pay for it
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram



7. Description

1. The Project Manager defines the Facilitator
2. The Project Manager sends the form *F-2a Define system mission, purpose, and objectives* to all the Stakeholders
3. The Stakeholders answer individually the form *F-2a Define system mission, purpose, and objectives*
4. The Stakeholders send their contributions (form *F-2a Define system mission, purpose and objectives* completed) to the Facilitator
5. The Facilitator integrates all Stakeholders answers in the form *F-2b Collaborative Session to define system mission, purpose and objectives*, identifying every Stakeholder with her/his contribution, it is suggested to use a different color for each Participant

   The form F-2b Collaborative Session to define system mission, purpose and objectives is suggested to be in a large format, with craft paper and post-it that allow the ex-change of information during the collaborative session
6. The Facilitator reunites the Stakeholders in a collaborative session that may be virtual, and it is suggested to occurs in the stakeholder facilities

7. During the collaborative session, the Participants negotiate and make consensus of all the answers, and prioritize the scenarios in order to reflect the weighted importance of the various operational needs.
8. The Facilitator completes the form *F-2c System mission, purpose, and objectives* with the resulting negotiated and consensuses answers.
9. Once the form *F-2c System mission, purpose, and objectives* is verified, the authorized stakeholder validate it.

8. Documents

F - 2a	Define system mission, purpose, and objectives
F - 2b	Collaborative session to define system mission, purpose, and objectives
F - 2c	System mission, purpose and objectives

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-3 Elicit Stakeholder Needs

DREAM	Work Instruction **Elicit Stakeholder Needs**	ID Work Instruction W – 3 Version 001

1. Objective To elicit, capture, and consolidate the stakeholder needs, expectations, objectives, and constraints guided by Stakeholders needs type classification
To assign a formal identifier to Stakeholder Needs

2. Responsible The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope This W may be applied to New System Development projects

4. Definitions **Rational:** Argument that provides the justification for the selection of an engineering element.
Requirement taxonomy:

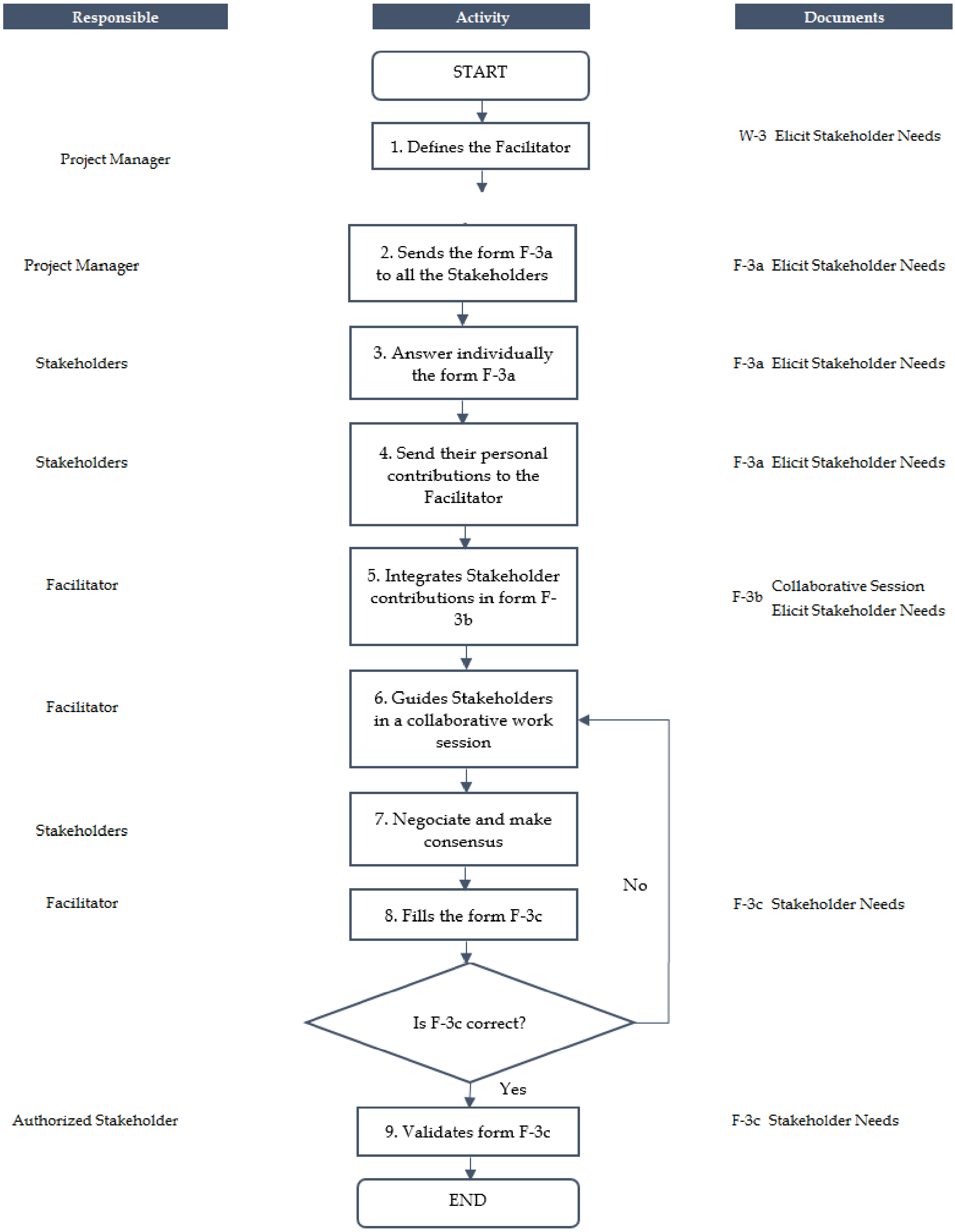
Requirement Taxonomy

- 1. Functional Requirements
- 2. Effectiveness/Performance Requirements
- 3. Interface Requirements
  - 3.1 Functional Interface Requirements
  - 3.2 Physical Interface Requirements
- 4. Utilization or Operational Requirements
  - 4.1 Modes and Scenarios
    - 4.1.1 Operational Modes
    - 4.1.2 Operational Scenarios
    - 4.1.3 Incident Modes
    - 4.1.4 Incident Scenarios
  - 4.2 Environmental Conditions Requirements
  - 4.3 Resources or used means Requirements
  - 4.4 Maintenance or Logistic Support Requirements
  - 4.5 Ergonomics and Human Factors Requirements
  - 4.6 Transportation and Storage Requirements
  - 4.7 Documentation Requirements
  - 4.8 Dependability and Robustness Requirements
    - 4.8.1 Immunity Requirements
    - 4.8.2 Innocuity Requirements
    - 4.8.3 Availability Requirements
    - 4.8.4 Survivability Requirements
    - 4.8.5 Reliability Requirements
    - 4.8.6 Mantainability Requirements
    - 4.8.7 Modularity Requirements
    - 4.8.8 Interoperability Requirements
    - 4.8.9 Reconfigurability Requirements
    - 4.8.10 Modifiability Requirements
    - 4.8.11 Extensibility Requirements
    - 4.8.12 Scalability Requirements
    - 4.8.13 Flexibility Requirements
    - 4.8.14 Adaptability Requirements
    - 4.8.15 Versatility Requirements
    - 4.8.16 Evolvability Requirements
    - 4.8.17 Changeability Requirements
    - 4.8.18 Information Availability Requirements
    - 4.8.19 Information Integrity Requirements
    - 4.8.20 Information Confidentialy Requirements
- 5. Constraints
  - 5.1 Physical Constraints
  - 5.2 Implementation Constraints
  - 5.3 Extension Contraints
  - 5.4 Transfer for Use Constraints
  - 5.5 Disposal Constraints
    - 5.5.1 Destruction Constraints
    - 5.5.2 Storage Constraints
  - 5.6 Constraints of Cost and Delivery of the product or service
  - 5.7 Manufacturing Constraints
  - 5.8 Regulation Constraints
  - 5.9 Standardisation Constraints
  - 5.10 Recovery Constraints
    - 5.10.1 Reuse Constraints
    - 5.10.2 Refurbish Constraints
    - 5.10.3 Remanufacture Constraints
    - 5.10.4 Recycle Constraints
  - 5.11 Other Quality Constraints
- 6. Forensic Requirements
  - 6.1 Detection of security incidents Requirements
  - 6.2 Assist with investigation of security incidents Requirements
    - 6.2.1 Availability of data for forensic investigators Requirements
    - 6.2.2 Collection of forensic data Requirements
    - 6.2.3 Secure storage of forensic data Requirements
    - 6.2.4 Prevent tampering of forensic data Requirements
    - 6.2.5 Examination of forensic data Requirements
  - 6.3 Eradication of security incidents Requirements
  - 6.4 Recovery from security incidents Requirements
- 7. Validation Requirements

Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram



7. Description

1. The Project Manager defines the Facilitator
2. The Project Manager sends the form *F-3a Elicit Stakeholder Needs* to all the Stakeholders
3. The Stakeholders answer individually the form *F-3a Elicit Stakeholder Needs*. It is possible that the Stakeholder does not have contributions to all the questions, or maybe more columns should be added to the form.
4. The Stakeholders send their contributions (form *F-3a Elicit Stakeholder Needs*) to the Facilitator
5. The Facilitator integrates all Stakeholders needs in one form *F-3b Collaborative Session Elicit Stakeholder Needs*, identifying every contribution with the Stakeholder who proposes it and its rationale; it is suggested to use colors to identify each Stakeholder contribution.

The form *F-3b Collaborative Session Elicit Stakeholder Needs* is suggested to be in a large format, with craft paper and post-it that allow the ex-change of information during the collaborative session

6. The Facilitator reunites the Stakeholders in a collaborative session that may be virtual, and it is suggested to occurs in the stakeholder facilities
7. During the collaborative session, the Participants negotiate and make consensus of all the needs
8. The Facilitator completes the form *F-3c Stakeholder Needs* including in every single record: the author Stakeholder and the rationale of the need and the formal identifier is defined for each Stakeholder Need as following:

SkN - XYZ – a

Where:
SkN: Stakeholder need
X: Number related to system requirement type classification group
Y: Number related to system requirement type classification sub-group (if available)
Z: Number related to system requirement type classification sub-sub-group (if available)
a: Consecutive number of stakeholder need

For example, SkN - 4.2.0 - 1 is the formal identifier of the stakeholder need (SkN) of environmental conditions (4.2) and is the first need of this type (1)

9. Once the form *F-3c Stakeholder Needs* is verified, the authorized stakeholder validate it.

8. Documents

F - 3a Elicit Stakeholder Needs
F - 3b Collaborative Session to Elicit Stakeholder Needs
F - 3c Stakeholder Needs

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-4 Stakeholder Needs feasibility and consistency

Work Instruction **Stakeholder Needs feasibility and consistency**	ID Work Instruction W – 4 Version 001

1. Objective To determine the feasibility and consistency of the set of stakeholder needs

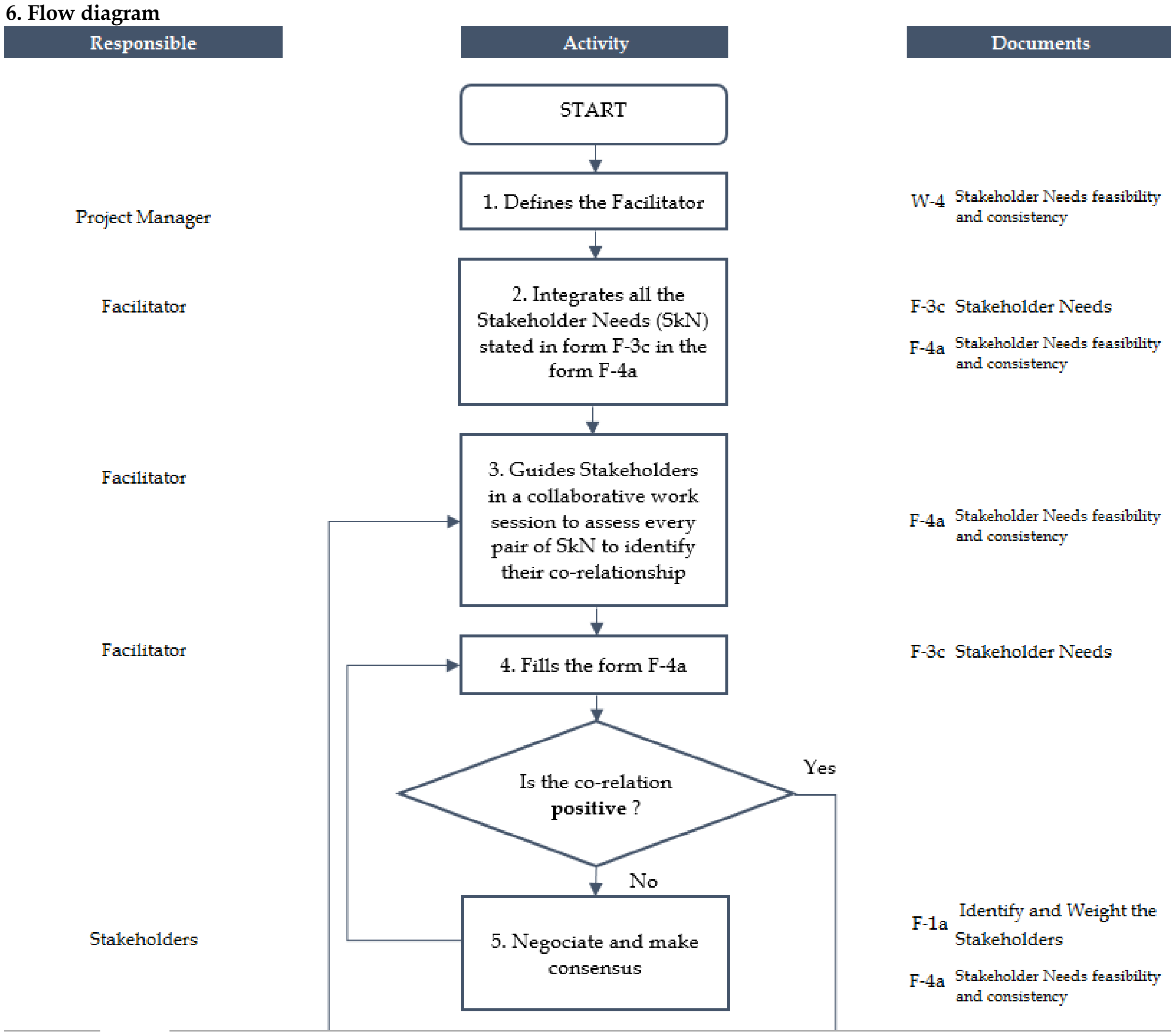
2. Responsible The Project Manager is responsible for the correct application of this Work Instruction (W)

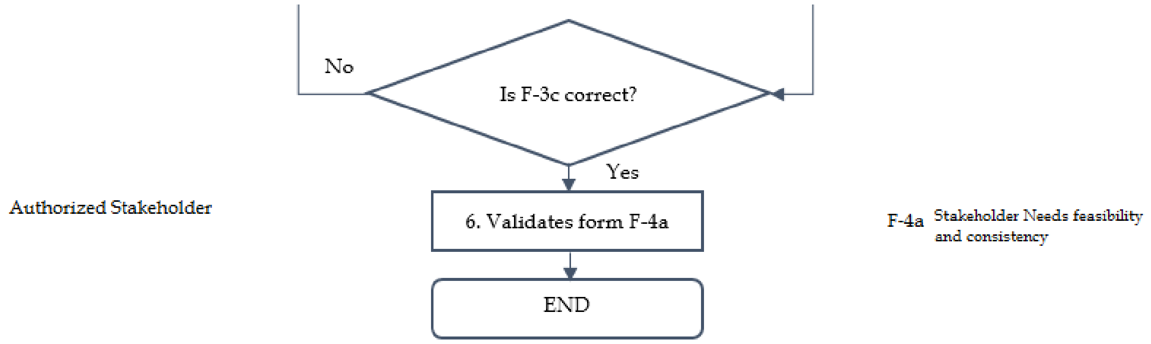
3. Scope This W may be applied to New System Development projects

4. Definitions
Set of Stakeholder Needs: The completeness of Stakeholder Needs (SkN)
Feasibility: Must be achievable using existent technologies and manufacturing; can be satisfied by a solution that is feasible within life-cycle
Consistency: Does not have individual requirements (needs) contradictories
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities
Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram





7. Description

1. The Project Manager defines the Facilitator
2. The facilitator integrates all SkN stated in form *F-3c Stakeholder Needs* in the form *F-4a Stakeholder Needs feasibility and consistency*
3. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session every pair of SkN are analyzed, to identify how, how strong and in what sense (positive or negative) they are co-related
4. The Facilitator fills the left down colored section of form *F-4a Stakeholder Needs feasibility and consistency* according to the following code:

   ++ Strong positive: direct and strong co-relation between the SkN
   \+ Medium positive: direct and weak co-relation between the SkN
   \- Medium negative: indirect and weak co-relation between the SkN
   -- Strong negative: indirect and strong co-relation between the SkN
   (nothing) If there is no co-relation between the SkN

   Once all relationships are defined, the feasibility and consistency of every pair must be assessed. This way it is assured that the system will be feasible and consistent.
5. When a pair of SkN is not feasible nor consistent, the participants must negotiate until making consensus.
   It is suggested that the stakeholders high weighted make the decisions. Use *F-1a Identify and Weight the Stakeholders*
   It is recommended the application of the Win-Win negotiation model through the form *F-4b Stakeholder Negotiation.*
   Instructions to fill the form:

5.1 Enter the Win conditions (stakeholder needs with negative -strong or medium- co-relationship)
5.2 Enter the Issues that summarize the involved conflict among the Win conditions
5.3 Prepare the candidate Options addressing every issue. The stakeholders evaluate the options to converge on a mutually satisfactory option
5.4 This Option is formally expressed in the Agreement schema
5.5 The facilitator verify the correct content of the form *F-4b Stakeholder Negotiation*
5.6 The authorized stakeholder validates it.

6. The facilitator, with the aid of the participants, verify the correct content of the form *F-4a Stakeholder Needs feasibility and consistency*. Once the form *F-4a Stakeholder Needs feasibility and consistency* is verified, the authorized stakeholder validate it.

8. Documents

F - 1a	Identify and Weight the Stakeholders
F - 3c	Stakeholder Needs
F - 4a	Elicit Stakeholder Needs
F - 4b	Stakeholder Negotiation

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-5 Prioritize Stakeholder Needs

DREAM	**Work Instruction** **Prioritize Stakeholder Needs**	ID Work Instruction W – 5 Version 001

1. Objective

To prioritize the retained stakeholder needs
To combine the prioritized stakeholder needs with stakeholder's weights
To determine the relative weight among stakeholder needs

2. Responsible

The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope

This W may be applied to New System Development projects

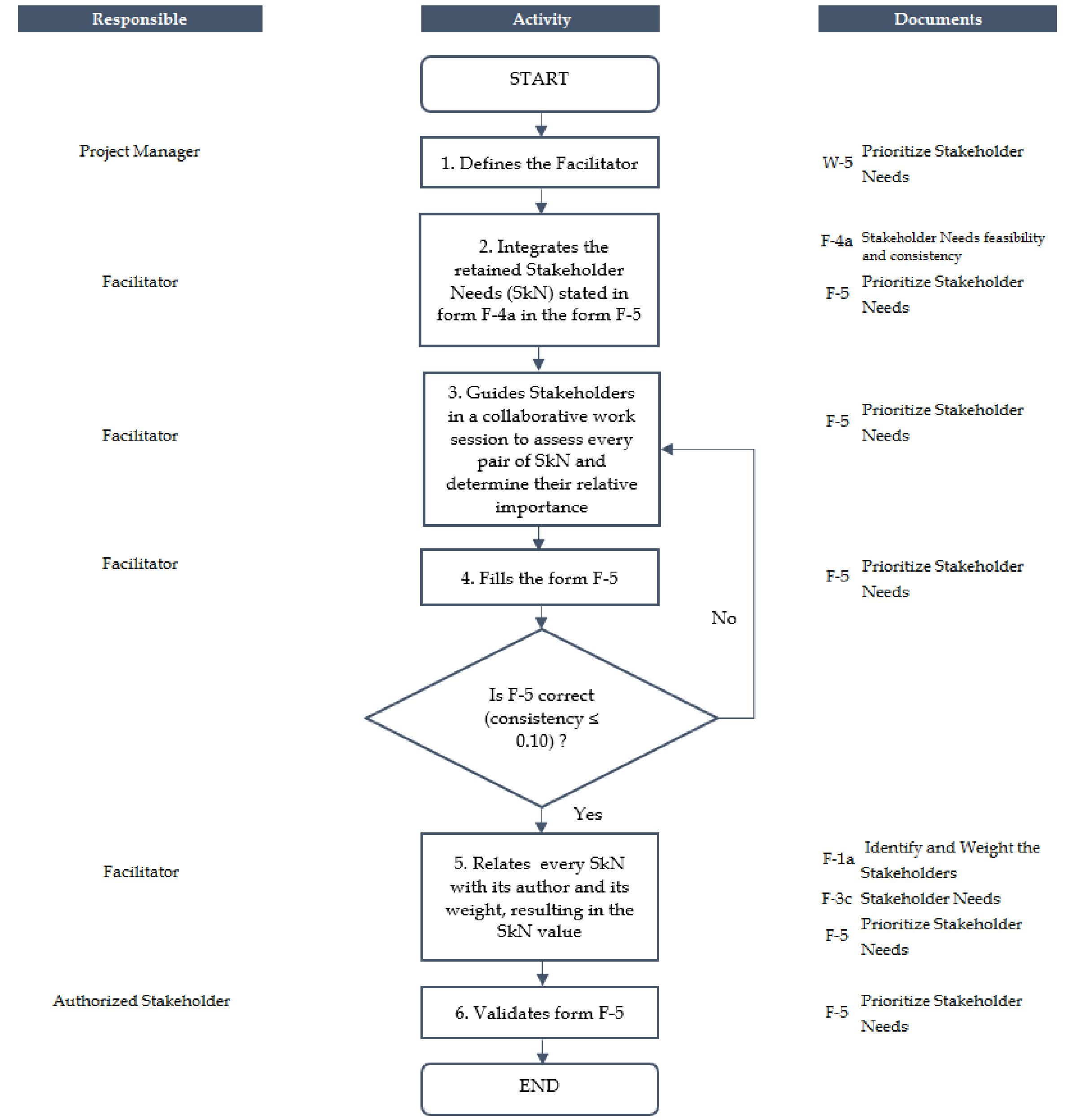
4. Definitions

Prioritize: Determine the order for dealing with (a series of items or tasks) according to their relative importance.
AHP: Analytical Hierarchy Process to detect inconsistencies in priority assignation.
Matrix: Mathematics: A rectangular array of quantities or expressions in rows and columns that is treated as a single entity and manipulated according to particular rules.
Vector: Mathematics: A matrix with one row or one column.
Consistency: Does not have individual requirements (needs) contradiction
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram



7. Description

1. The Project Manager defines the Facilitator
2. The facilitator integrates the retained Stakeholders needs stated in form *F-4a Stakeholder Needs feasibility and consistency* in form *F - 5 Prioritize Stakeholder Needs*
3. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session every pair of SkN are analyzed, and the participants discuss which one is more important than the other according to the values contained in Table 1.

	Intensity Scale	
Less important than	Extremely less important	1/9
		1/8
	Very strong less important	1/7
		1/6
	Strongly less important	1/5
		1/4
	Moderately less important	1/3
		1/2
	Equal Importance	1
More important than		2
	Moderately more important	3
		4
	Strongly more important	5
		6
	Very strong more important	7
		8
	Extremely more important	9

Table 1. Comparative judgment table (Danesh & Ryan, 2015)

Example:
If SkN _ 1.0.0 _ 1 is moderately more important than SkN _ 1.0.0 _ 2, then the value to choose is 3
If SkN _ 1.0.0 _ 1 is very strong less important than SkN _ 1.0.0 _ 2, then the value to choose is 1/7

4. Once the participants have made consensus of every pair of need importance relationships, the Facilitator fills the left down colored section of MATRIX A, included in form *F - 5 Prioritize Stakeholder Needs*
   Automatically a second normalized MATRIX B is calculated
   Finally, the SkN will be listed, including the prioritized value of each one

It is highly recommended that the Facilitator calculates the consistency among the stakeholder responses when filling the pairwise comparison matrix (MATRIX A). Consistency analysis is done through the calculation of the consistency ratio (CR). If this value is bigger than 0.10 it means that there are inconsistencies among the judgments. In that case, the value assignment should be repeated until the consistency value is minor or equal to 0.10

5. The facilitator relates every SkN with its author and its weight. This information can be retrieved from the forms F-1a Identify and Weight the Stakeholders and F-3c Stakeholder Needs.

Automatically is calculated the Stakeholder Need Value and the Relative Stakeholder Need Value, that shows which needs are the most valuable that the system should meet.
NOTE: the Relative Stakeholder Need Value is a very interesting data; in later stages, when the system is built and ready to be validated, this data will help to quantify if the system meets or not the SkN

6. Once the Facilitator verifies the correct filling of the form F-5 Prioritize Stakeholder Needs, the authorized stakeholder validate it

8. Documents

F - 1a	Identify and Weight the Stakeholders
F - 3c	Stakeholder Needs
F - 4a	Elicit Stakeholder Needs
F - 5	Prioritize Stakeholder Needs

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-6 List of Stakeholder Requirements

DREAM	Work Instruction **List of Stakeholder Requirements**	ID Work Instruction W – 6 Version 001

1. Objective
To translate stakeholder needs (SkN) into stakeholder requirements (SkR)
To classify stakeholder requirements by type
To define derived and decomposed stakeholder requirements as needed
To assign a formal identifier to stakeholder requirements

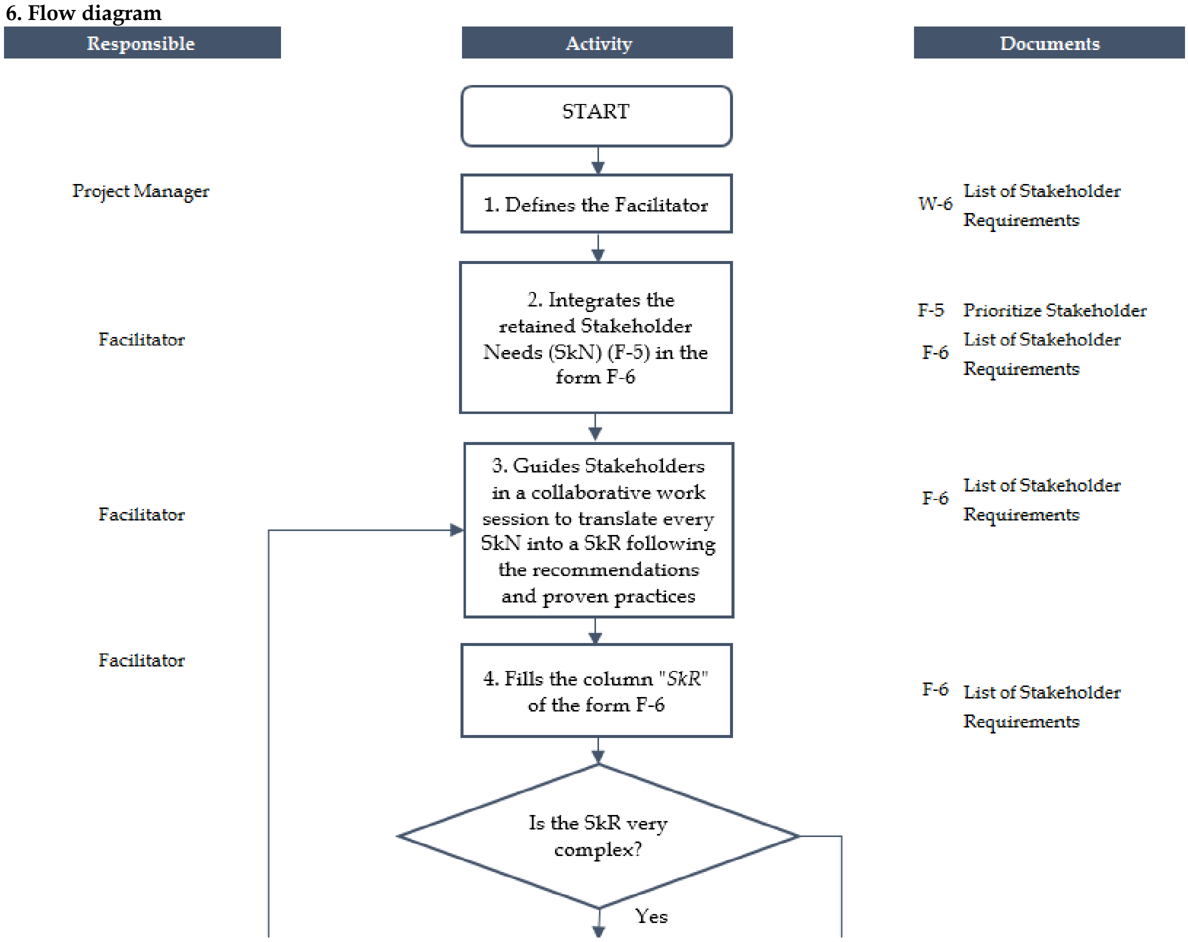
2. Responsible
The Project Manager is responsible for the correct application of this Work Instruction (W)

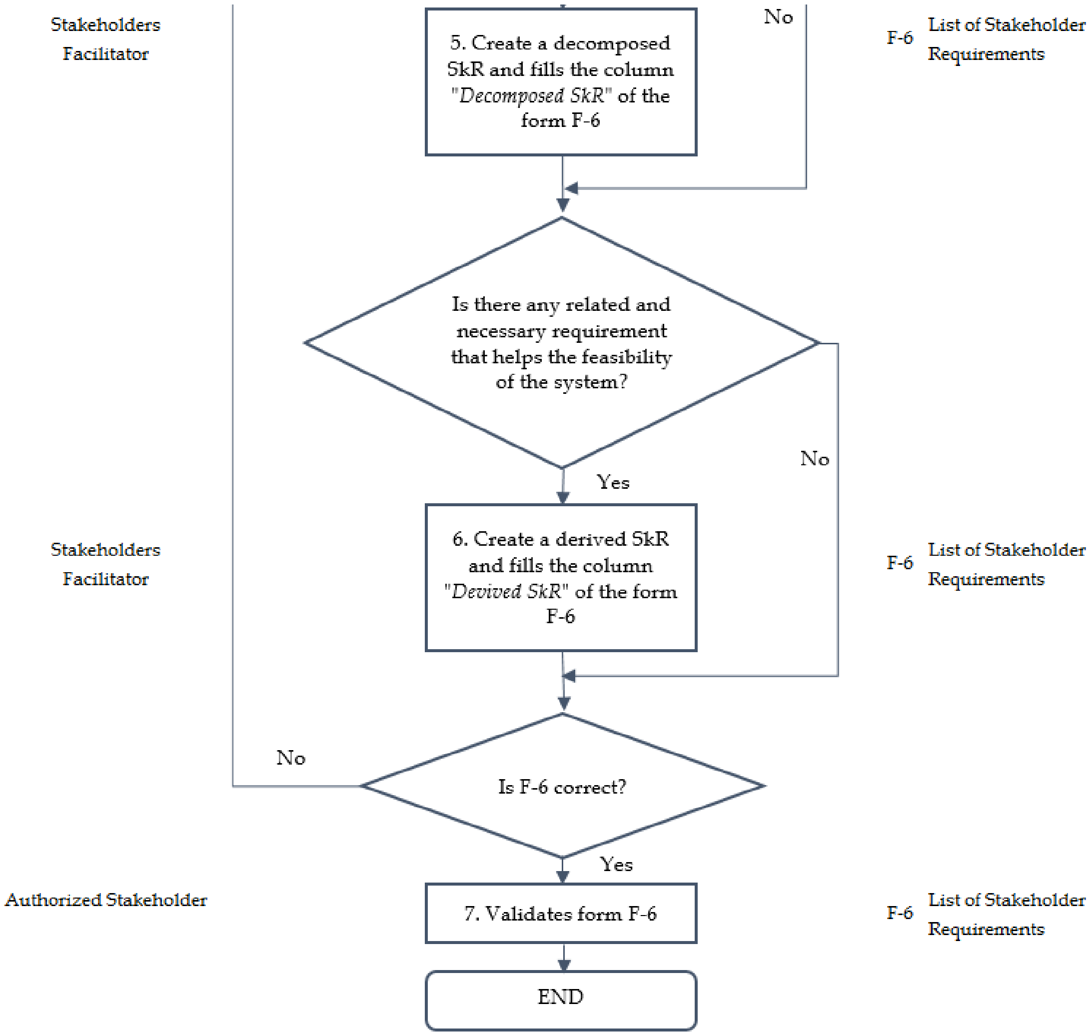
3. Scope
This W may be applied to New System Development projects

4. Definitions
Decomposed requirement: Those requirements that come from complex requirements and that need to be split in two or more for their correct treatment.
Derived requirement: Those requirements that are necessary to the feasibility of the system, without them the system will not be able to exist
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities
Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram





7. Description

1. The Project Manager defines the Facilitator
2. The facilitator integrates the retained SkN stated in the last table of the form *F-5 Prioritize Stakeholder Needs* in the form *F-6 List of Stakeholder Needs*
3. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session, the Facilitator guides the participants through the analysis of each SkN.

The suggested pattern for writing requirements is the following:

System/system component (subject) + Necessity (modal verb) +
Function/behavior/characteristics (verb phrase) + Condition (adjunct)

Examples:

The system shall provide first aid for injured people
The system shall be built using an existing industrial rolling base
The system shall be available 70% of the time

Recommendations:

X Avoid vague and general terms
X Avoid unbounded or ambiguous terms like:
- x Superlatives such as best or most
- x Subjective language such as user-friendly, easy to use, cost-effective
- x Vague pronouns such as it, this, that
- x Ambiguous adverbs and adjectives such as: almost, always, significant, minimal
- x Open-ended, non-verifiable terms such as: provide support, but not limited to, etc.
- x Comparative phrases such as: better than, higher quality
- x Loopholes such as: if possible, as appropriate, as applicable
- x Incomplete references like not specifying the reference with its date and version number
- x Negative statements

Proven practices:

✓ Involve stakeholders early in the analysis
✓ Presence of rationale for each stakeholder requirement
✓ Analyze sources of stakeholders requirements before starting the definition of the system requirements
✓ Use of modeling techniques

✓ Use of requirements managing tool to trace linkages and to record the source of each stakeholder requirement
✓ Model the system of interest (SoI) from a higher-level system to identify and define its context (services, functional and physical interfaces, etc.)
✓ Use language and synonym dictionaries. Semantics is the key to the correct expression of requirements
✓ When talking about a complex requirement, write complementary requirements that restrict the possible deviated interpretation of the original requirement.

The requirement statement should fulfill the next quality characteristics:
Mature: Is the expression of the SkR close to Stakeholders expectations?
Accurate: Did the Stakeholder express their expectations with precision?
Feasible: Was the SkR feasibility assessed through identified operational concepts?
Appropriate to level: Is the SkR appropriate to the level at it is stated?
Complete: Is the SkR explained enough?
Conforming: Is the SkR conform to a standard formal structure?
Verifiable: Can be verified that the system meets or possesses the SkR?
Necessary: Should the system be able to function in the desired way with this SkR?
Singular: Is not the SkR a combination of two or more requirements?
Correct: Is this a SKR that will result in the desired system performance?
Unambiguous: Is there only one interpretation of the SkR?
Implementation free: Does the SkR states what is required instead of saying how?
Consistent: Is free of conflicts with other requirements?

4. Once there is consensus, the Facilitator should complete the second column "*Stakeholder Requirement SkR*" of form *F - 6 List of Stakeholder Requirements*

Once finished, the facilitator should guide the participants through the analysis of each individual SkR:

a) Is the SkR very complex? If the SkR is split, it would have a better and correct treatment?

5. If the answers are "yes", then it may be suitable to create decomposed SkR as needed

b) Is there any related and necessary requirement that helps the feasibility of the system?

6. If the answers are "yes", then it may be suitable to create derived SkR

If needed, the facilitator continues filling the following column "*Derived or Decomposed SkR*" in form *F - 6 List of Stakeholder Requirements*.
The facilitator, while filling the form, is constructing the formal identifier of the SkR
The facilitator fills the form *F-6 List of Stakeholder Requirements*, and together, with aid of the participants, verify the correct content of the form. The following question should help to synthesize this list:

Are there duplicated system requirements?

7. Once the form F-6 List of Stakeholder Requirements is verified, the authorized stakeholder validate it.

8. Documents

F - 5 Prioritize Stakeholder Needs
F - 6 List of Stakeholder Requirements

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-7 Stakeholder Requirements

DREAM	Work Instruction **Stakeholder Requirements**	ID Work Instruction W – 7 Version 001

1. Objective
To assign Stakeholder Requirement (SkR) attributes
To identify the related Stakeholders of each SkR
To identify the Stakeholders responsible for verification and validation (V&V) activities
To verify the quality of the SkR statement
To verify and validate (V&V) objectively the SkR
To identify and mitigate the potential risk that could be generated by SkR
To identify the SkR upward traceability
To identify the SkR bi-directional traceability and its co-relationships

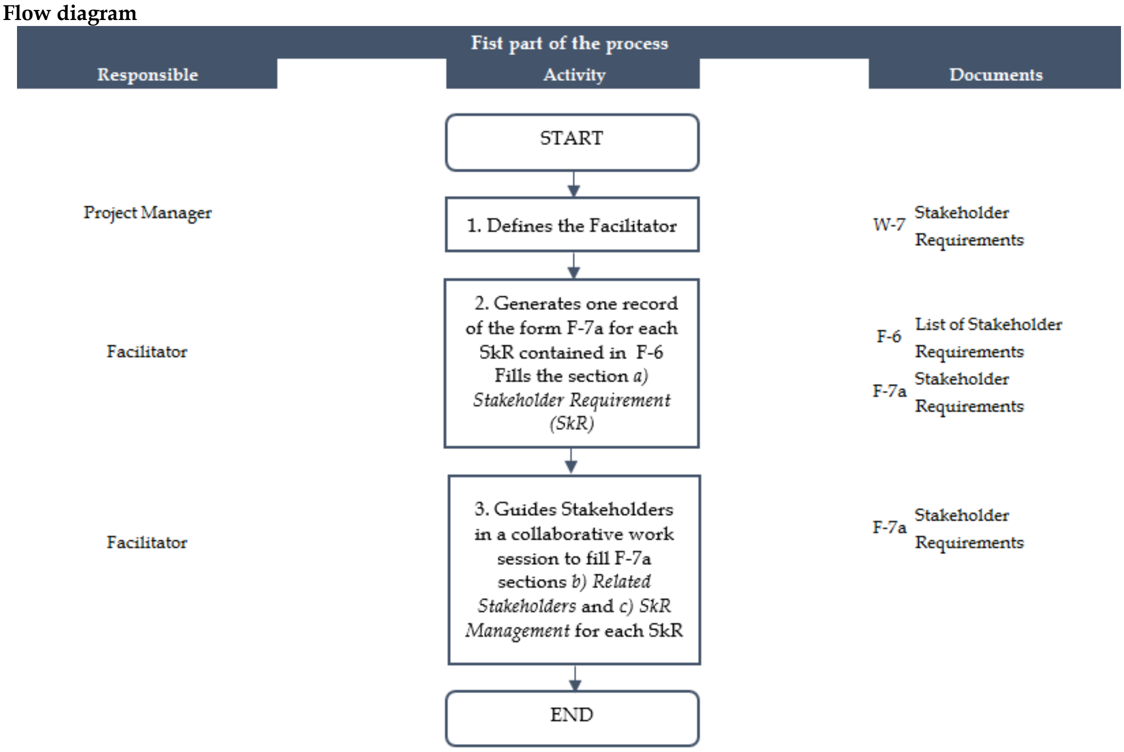
2. Responsible
The Project Manager is responsible for the correct application of this Work Instruction (W)

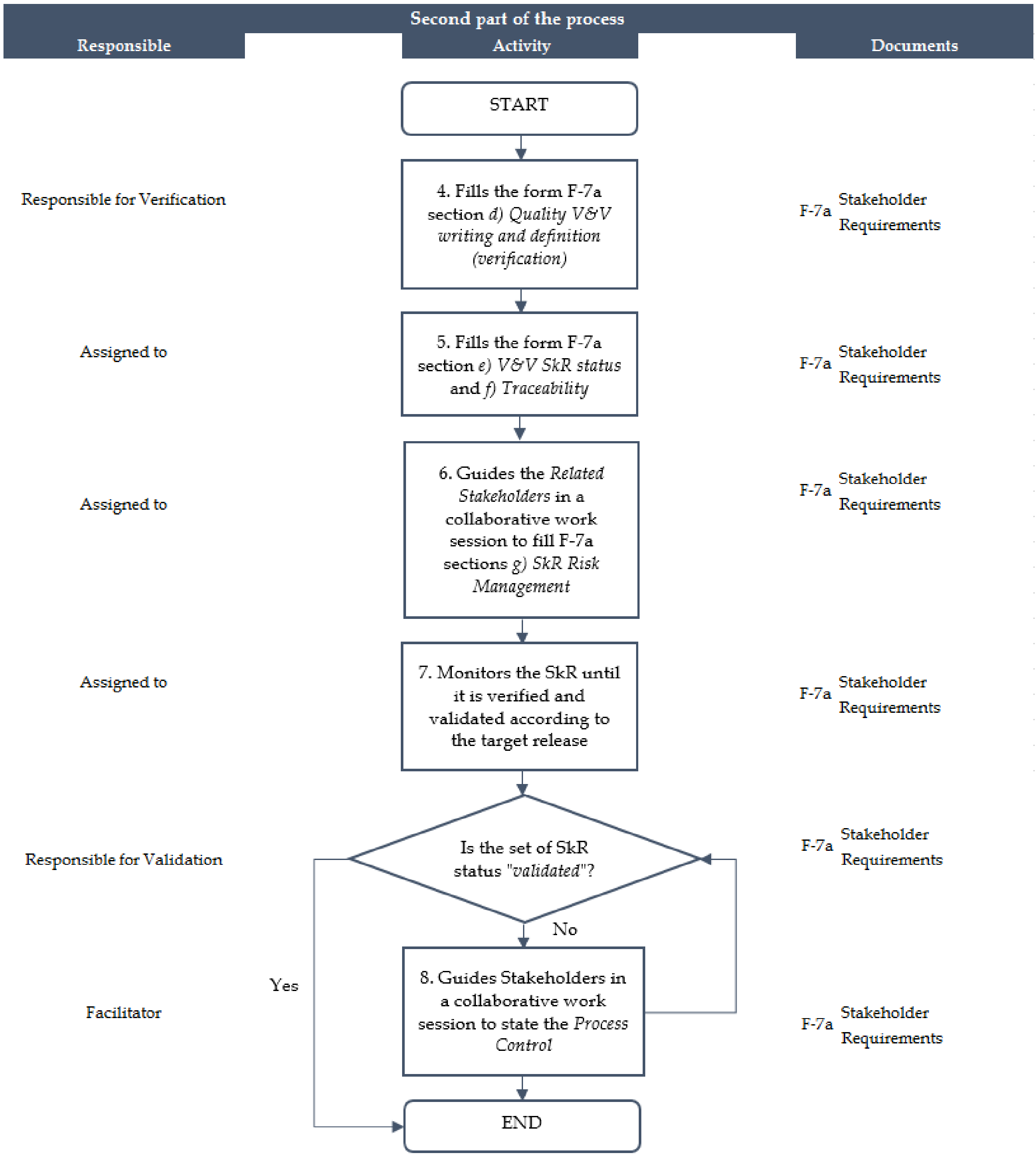
3. Scope
This W may be applied to New System Development projects

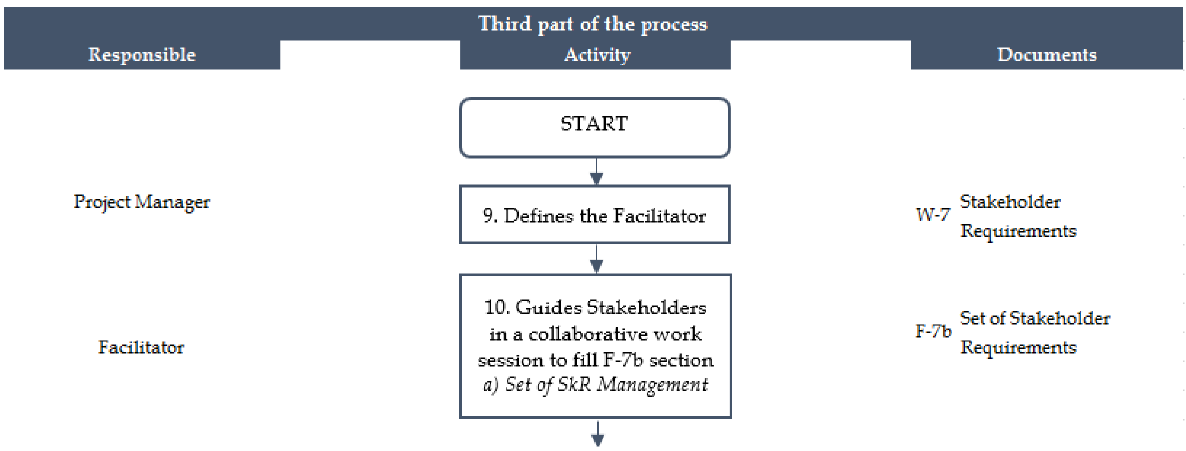
4. Definitions
Attribute: additional information included with a requirement statement, which is used to aid in the management of that requirement.
Verify: To verify a stakeholder requirement or a system requirement is to check the application of syntactic and grammatical rules, and characteristics defined in the stakeholder requirements definition process, and the system requirements definition process such as; necessity, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, and verifiable.
Validate: To validate a stakeholder requirement, make sure its content is justified and relevant to stakeholders' expectations, complete and expressed in the language of the customer or end user.
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

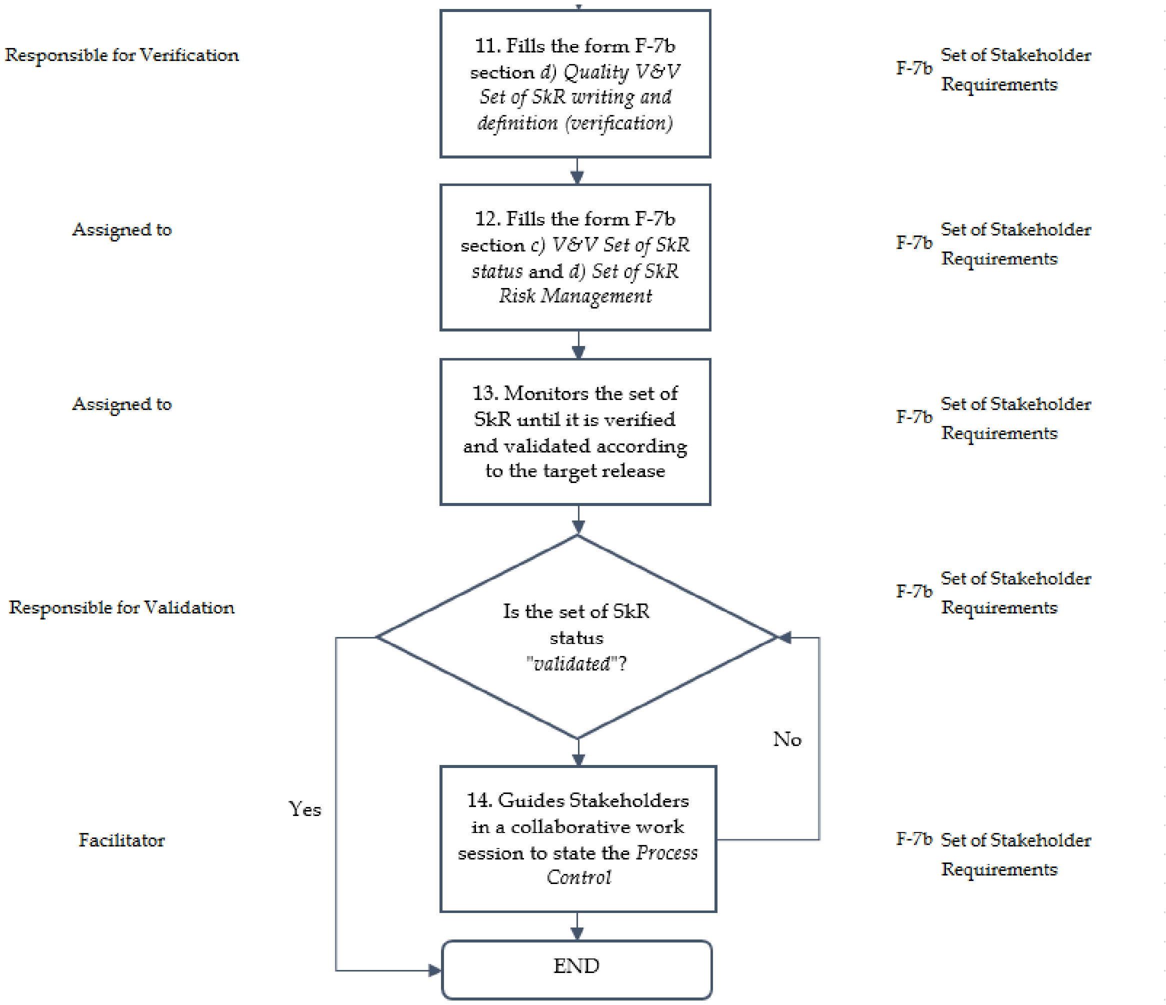
5. Generalities
Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram




Second part of the process
Responsible
Activity
Documents
START
Responsible for Verification
4. Fills the form F-7a section *d) Quality V&V writing and definition (verification)*
F-7a
Stakeholder Requirements
Assigned to
5. Fills the form F-7a section *e) V&V SkR status* and *f) Traceability*
F-7a
Stakeholder Requirements
Assigned to
6. Guides the *Related Stakeholders* in a collaborative work session to fill F-7a sections *g) SkR Risk Management*
F-7a
Stakeholder Requirements
Assigned to
7. Monitors the SkR until it is verified and validated according to the target release
F-7a
Stakeholder Requirements
Responsible for Validation
Is the set of SkR status "*validated*"?
F-7a
Stakeholder Requirements
No
Yes
Facilitator
8. Guides Stakeholders in a collaborative work session to state the *Process Control*
F-7a
Stakeholder Requirements
END


Third part of the process
Responsible
Activity
Documents
START
Project Manager
9. Defines the Facilitator
W-7
Stakeholder Requirements
Facilitator
10. Guides Stakeholders in a collaborative work session to fill F-7b section *a) Set of SkR Management*
F-7b
Set of Stakeholder Requirements



7. Description

The first part of the process

1. The Project Manager defines the Facilitator
2. The Facilitator generates one record of form F - 7a Stakeholder Requirement for each SkR contained in *F - 6 List of Stakeholder Requirements*, and fills the section *a) Stakeholder Requirement (SkR)*
3. The facilitator establishes a collaborative session and invite the Stakeholders; during the collaborative session the Facilitator guides the participants through the analysis of each SkR to fill sections *b) Related Stakeholders* in every stage of the system life cycle (included the Stakeholder authors of the correlated SkNs); and *c) SkR management*: "Assigned to" (stakeholder responsible of this SkR), "Responsible for verification", and "Responsible for validation"

NOTE: To reduce wasting time in the process, it is suggested that the collaborative sessions for the *Second part of the process* should be performed only among the *Related Stakeholders* of each SkR.

The second part of the process

4. The "Responsible for Verification" and the "Responsible for Validation" fill the section *d) Quality: V&V writing and definition*.
5. The "Assigned to" fills the section *e) V&V SkR status and f) Traceability*
6. After discussion and consensus among the "Related Stakeholders", the "Assigned to" fills the section *g) SkR Risk Management* according to the following information:

Likelihood		
Score	Likelihood of Occurrence (p)	
5	Near certainty	$p > 80\%$
4	Highly Likely	$60\% < p \leq 80\%$
3	Likely	$40\% < p \leq 60\%$
2	Low likelihood	$20\% < p \leq 40\%$
1	Not likely	$p \leq 20\%$

CONSEQUENCE					
Score	1	2	3	4	5
Performance	Minimal consequence to objectives/goals	Minor consequence to objectives/goals	Unable to achieve a particular objective/goal, but remaining objective goals represent better than minimum success or outcome	Unable to achieve multiple objectives/goals but minimum success can still be achieved or claimed	Unable to achieve objectives/goals such that minimum success cannot be achieved or claimed
Safety Human	Discomfort or nuisance	First aid event per OSHA criteria	No lost time injury or illness per OSHA criteria	Lost time injury or illness per OSHA criteria	Loss of life
Asset	Minimal consequence: asset has no sign of physical damage	Minor consequence: asset has cosmetic damage and is repairable	Minor consequence: asset is damaged but repairable	Major consequence asset is substantially damaged but repairable	Destroyed: asset is compromised, and un-repairable: a total loss
Schedule	Minimal consequence	Critical path is not slipped; total slack of slipped tasks will not impact critical path in less than 10 days	Critical path is not slipped; total slack of slipped tasks is within 10 days of impacting the critical path	Critical path slips	Critical path slips and one or more critical milestones or events cannot be met
Cost	Minimal consequence	Minor cost consequence. Cost variance ≤ 5% of total approved FY baseline	Cost consequence. Cost variance >5% but ≤ of total approved FY baseline	Cost consequence. Cost variance >10% but <15% of total approved FY baseline	Major cost consequence. Cost consequence. Cost variance >15% of total approved FY baseline

7. The "Assigned to" monitors the SkR until it is verified and validated according to the target release, and its status is "accepted"
8. The Facilitator guides the Stakeholders in a collaborative work session to state the Process Control, corrective actions due to non-conformities detection

The third part of the process

9. The Project Manager defines the Facilitator
10. The facilitator establishes a collaborative session and invites the Stakeholders; during the collaborative session the Facilitator guides the Stakeholders to fill the form *F-7b Set of Stakeholder Requirements* section *a) Set of SkR Management*
11. The "Responsible for Verification" fills the form *F-7b Set of Stakeholder Requirements* section *d) Quality V&V Set of SkR writing and definition (verification)*
12. The "Assigned to" fills the form *F-7b Set of Stakeholder Requirements* section *c) V&V Set of SkR status* and *d) Set of SkR Risk Management*
13. The "Assigned to" monitors the set of SkR until it is verified and validated according to the target release, and its status is "accepted"
14. The Facilitator guides the Stakeholders in a collaborative work session to state the *Process Control*, corrective actions due to non-conformities detection

8. Documents

F - 6	List of Stakeholder Requirements
F - 7a	Stakeholder Requirement
F - 7b	Set of Stakeholder Requirements

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-8 Analyze Stakeholder Requirements

DREAM	**Work Instruction** **Analyze Stakeholder Requirements**	ID Work Instruction W – 8 Version 001

1. Objective To analyze completeness and coherence of information

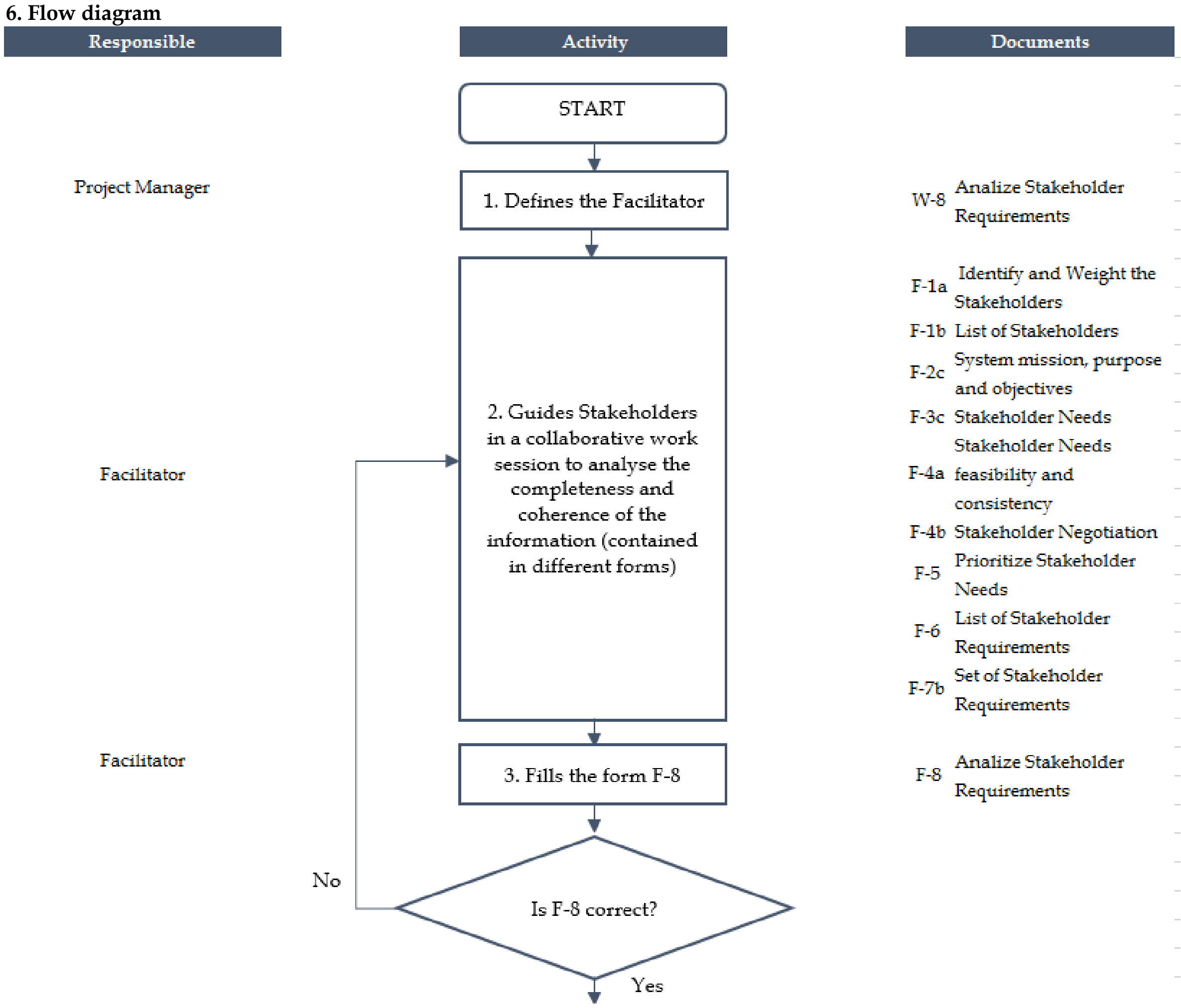
2. Responsible The Project Manager is responsible for the correct application of this Work Instruction (W)

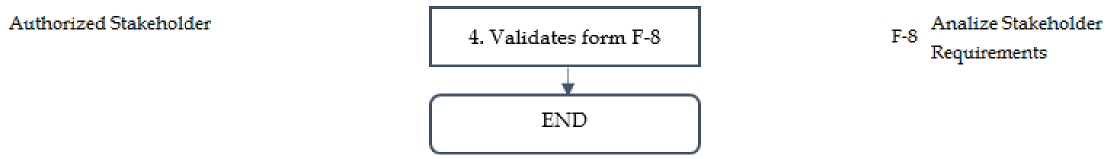
3. Scope This W may be applied to New System Development projects

4. Definitions **Facilitator:** Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram





7. Description

1. The Project Manager defines the Facilitator
2. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session the Facilitator guides the participants through the analysis and the completeness of the retrieved information:

2.1 Forms *F-1a Identify and Weight the Stakeholders* and *F-1b List of Stakeholders* will be useful when verifying that, as far as possible, the involved stakeholders are being identified and involved. In case that one or more stakeholders are identified, they should be added to forms *F-1a Identify and Weight the Stakeholders* and *F-1b List of Stakeholders* and involved in the project as soon as possible.

2.2 The form *F-2c System mission, purpose, and objectives* will be useful to verify that, as far as possible, the possible scenarios have been considered, as well the assumptions, what is the "*value*" for the stakeholders, and what is the "*success*" for the project, the system context of use, the functional and physical interfaces of the system, the input-output flows of material, energy and/or information, the necessary physical connections to carry the exchanged flows of material, energy and/or information. In case that some stakeholders are added into the project, they should be able to review and contribute with their ideas to complete the form *F-2c System mission, purpose, and objectives*; the stakeholder responsible of validation should agree.

2.3 In case that some stakeholders are added into the project, they should be able to review and contribute with their ideas to complete the form *F-3c Stakeholder Needs*; the stakeholder responsible for validation should agree.

2.4 *F-4a Stakeholder Needs feasibility and consistency*, *F-4b Stakeholder Negotiation*, and *F-5 Prioritize Stakeholder Needs* should be analyzed in case that one or more stakeholder requirements are added to the form *F-3c Stakeholder needs*; the stakeholder responsible of validation should agree.

2.5 The form *F-6 List of Stakeholder Requirements* and *F-7b Set of Stakeholder Requirements* will be useful to verify that, as far as possible, the stakeholder needs are identified and translated into stakeholder requirements, as well as the constraints; for example: the enterprise context, procedures, environmental conditions, programmatic constraints, available resources and available technologies, interfaces with existing systems, etc.

3. The Facilitator fills the form *F-8 Analyze Stakeholder Requirements*
4. Once the form *F-8 Analyze Stakeholder Requirements* is verified, the authorized stakeholder validate it.

8. Documents

F - 1a	Identify and Weight the Stakeholders
F - 1b	List of Stakeholders
F - 2c	System mission, purpose and objectives
F - 3c	List of Stakeholders
F - 4a	Stakeholder Needs feasibility and consistency
F - 4b	Stakeholder Negotiation
F - 5	Prioritize Stakeholder Needs
F - 6	List of Stakeholder Requirements
F - 7b	Set of Stakeholder Requirements
F - 8	Analyze Stakeholder Requirements

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-9 List of System Requirements

DREAM	**Work Instruction** **List of Stakeholder Requirements**	ID Work Instruction W – 9 Version 001

1. Objective

To translate stakeholder requirements (SkR) into system requirements (SyR)
To classify system requirements by type
To define derived and decomposed system requirements as needed
To assign a formal identifier to system requirements

2. Responsible

The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope

This W may be applied to New System Development projects

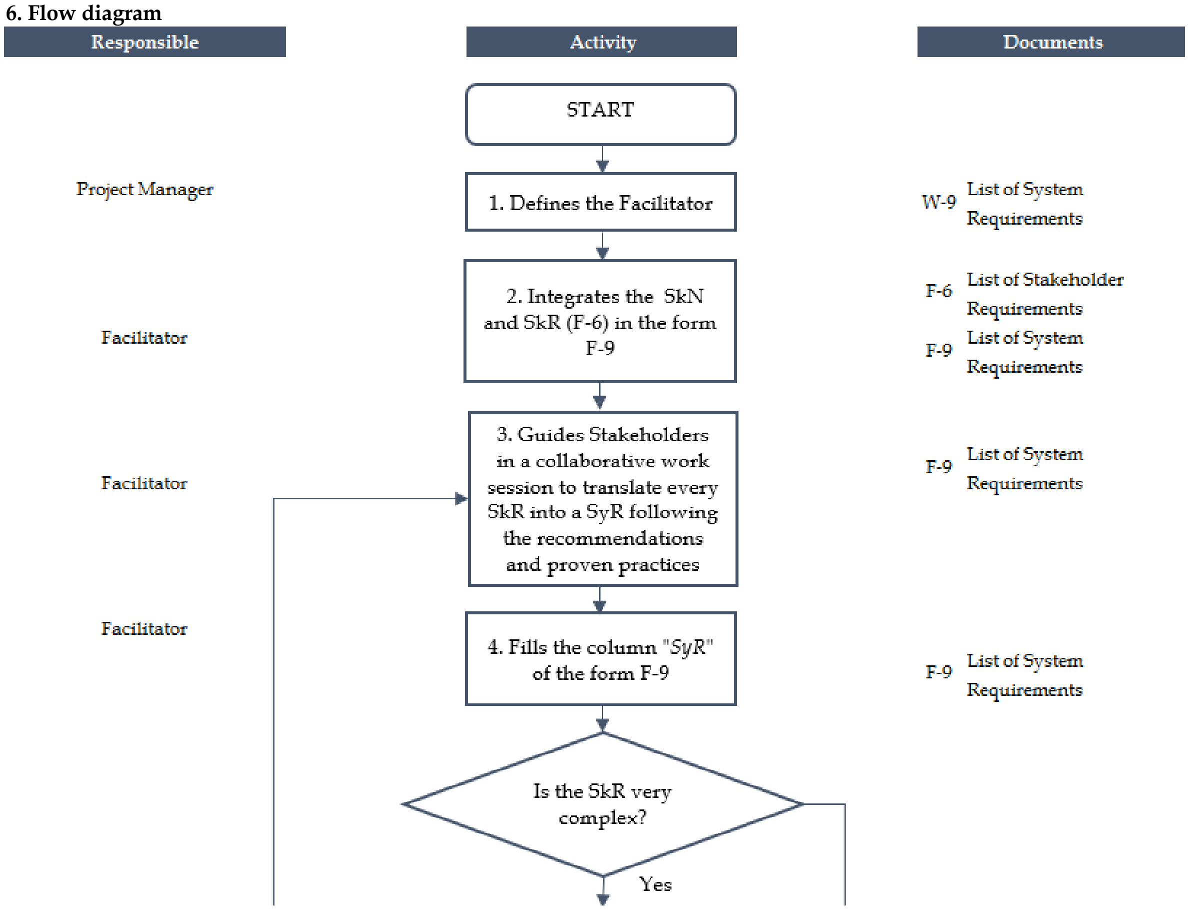
4. Definitions

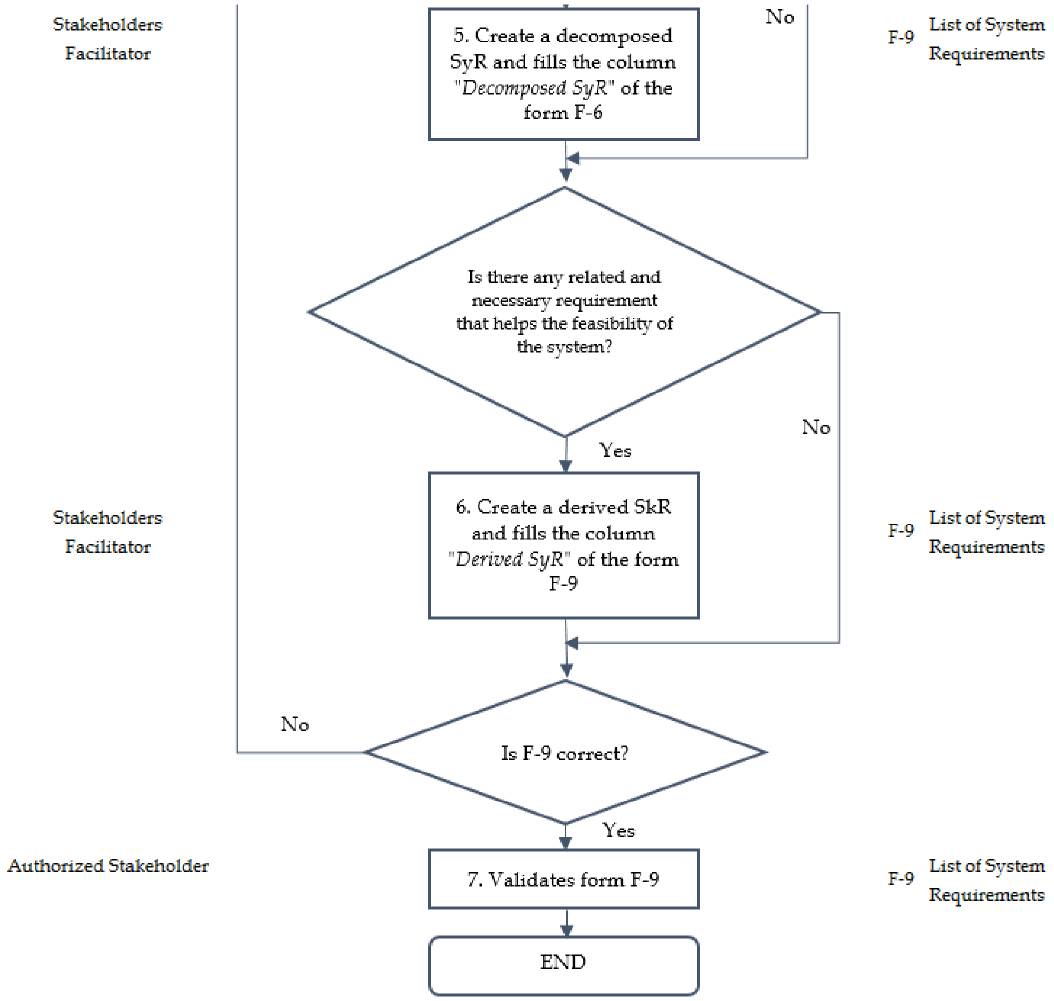
Decomposed requirement: Those requirements that come from complex requirements and that need to be split in two or more for their correct treatment
Derived requirement: Those requirements that are necessary to the feasibility of the system, without them the system will not be able to exist
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram





7. Description

1. The Project Manager defines the Facilitator
2. The facilitator integrates the information contained the form *F-6 List of Stakeholder Requirements* in the form *F-9 List of System Requirements*
3. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session, the Facilitator guides the participants through the analysis of each SkR.

The analysis consists of the reflection of every SkR statement to translate it into a SyR statement

The suggested pattern for writing requirements is the following:

System/system component (subject) + Necessity (modal verb) +
Function/behavior/characteristics (verb phrase) + Condition (adjunct)

Examples:

The system shall provide first aid for injured people
The system shall be built using an existing industrial rolling base
The system shall be available 70% of the time

Recommendations:

X Avoid vague and general terms
X Avoid unbounded or ambiguous terms like:
  - x Superlatives such as best or most
  - x Subjective language such as user-friendly, easy to use, cost-effective
  - x Vague pronouns such as it, this, that
  - x Ambiguous adverbs and adjectives such as: almost, always, significant, minimal
  - x Open-ended, non-verifiable terms such as: provide support, but not limited to, etc.
  - x Comparative phrases such as: better than, higher quality
  - x Loopholes such as: if possible, as appropriate, as applicable
  - x Incomplete references like not specifying the reference with its date and version number
  - x Negative statements

Proven practices:

✓ Involve stakeholders early in the analysis
✓ Presence of rationale for each stakeholder requirement
✓ Analyze sources of stakeholders requirements before starting the definition of the system requirements
✓ Use of modeling techniques

✓ Use of requirements managing tool to trace linkages and to record the source of each stakeholder requirement
✓ Model the system of interest (SoI) from a higher-level system to identify and define its context (services, functional and physical interfaces, etc.)
✓ Use language and synonym dictionaries. Semantics is the key to the correct expression of requirements
✓ When talking about a complex requirement, write complementary requirements that restrict the possible deviated interpretation of the original requirement.

The requirement statement should fulfill the next quality characteristics:
Feasible: Was the SyR feasibility assessed through identified operational concepts?
Appropriate to level: Is the SyR appropriate to the level at it is stated?
Complete: Is the SyR explained enough?
Conforming: Is the SyR conform to a standard formal structure?
Verifiable: Can be verified that the system meets or possesses the SyR?
Necessary: Should the system be able to function in the desired way with this SyR?
Singular: Is not the SyR a combination of two or more requirements?
Correct: Is this a SKR that will result in the desired system performance?
Unambiguous: Is there only one interpretation of the SyR?
Implementation free: Does the SyR states what is required instead of saying how?
Consistent: Is free of conflicts with other requirements?

4. Once there is consensus, the Facilitator should complete the second column "*System Requirement SyR*" of form *F - 9 List of System Requirements*

Once finished, the facilitator should guide the participants through the analysis of each individual SyR. The SyR are supposed to come from the stakeholder need elicitation process.

a) Does this requirement come from more complex requirements which are decomposed?

5. If the answers are "yes", then it may be suitable to create decomposed SyR as needed

b) Does this requirement come from more complex requirement which is decomposed & modeled?
c) Does this requirement come from architecture or design decisions?

6. If the answers are "yes", then it may be suitable to create derived SyR

If needed, the facilitator continues filling the following column "*Derived or Decomposed System Requirements (SyR)*" in form *F - 9 List of System Requirements*.
The facilitator, while filling the form, is constructing the formal identifier of the SyR
The facilitator fills the form *F-9 List of System Requirements*, and together, with aid of the participants, verify the correct content of the form. The following question should help to synthesize this list:
Are there duplicated system requirements?

7. Once the form *F-9 List of System Requirements* is verified, the authorized stakeholder validate it.

8. Documents

F - 6 List of Stakeholder Requirements
F - 9 List of System Requirements

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-10 Bi-directional Traceability at adjacent levels of the system hierarchy

DREAM	Work Instruction **Bi-directional Traceability at adjacent levels of the system hierarchy**	ID Work Instruction W – 10 Version 001

1. Objective To determine the feasibility and consistency of the set of system requirements (SyR)

2. Responsible The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope This W may be applied to New System Development projects

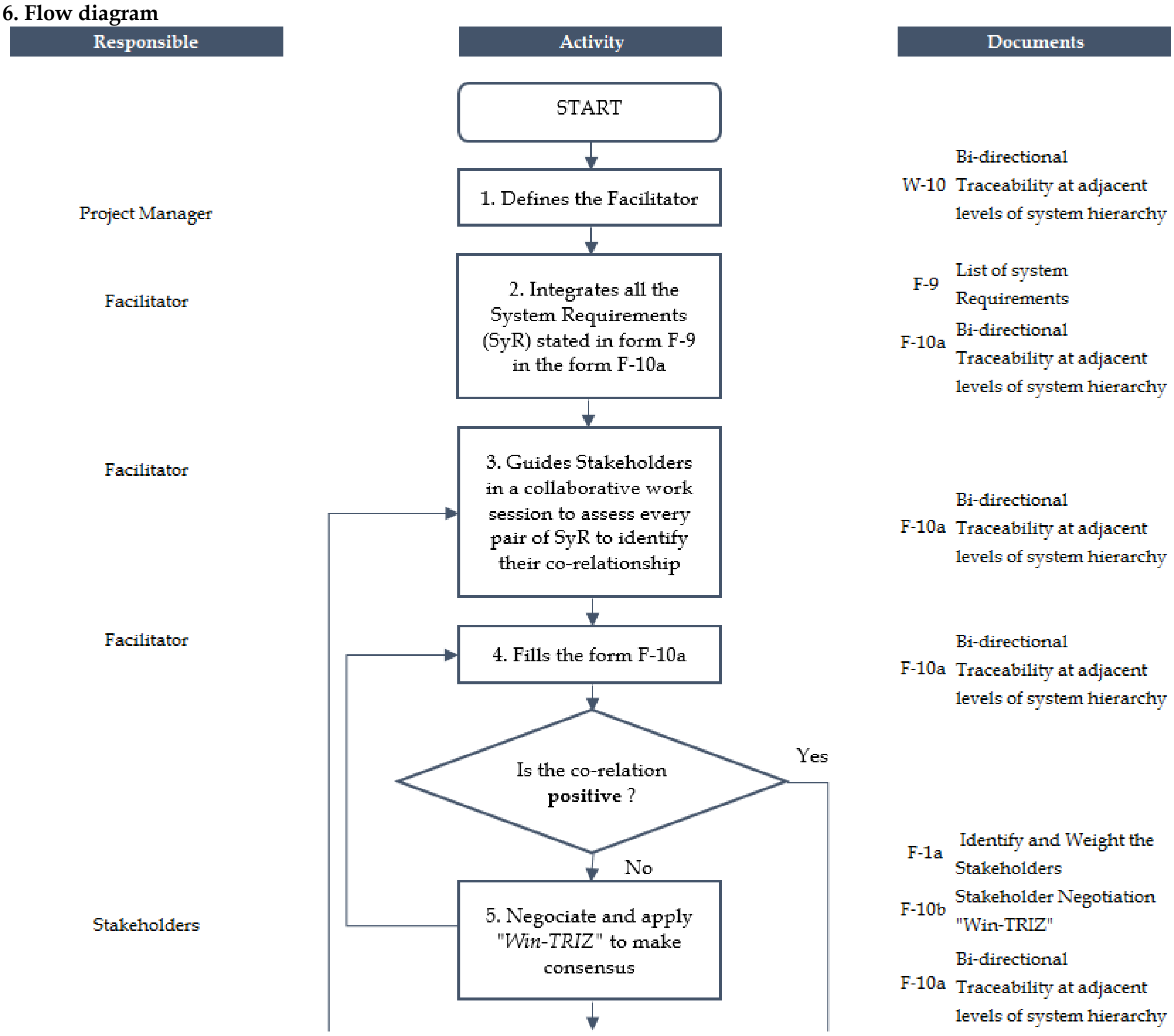
4. Definitions

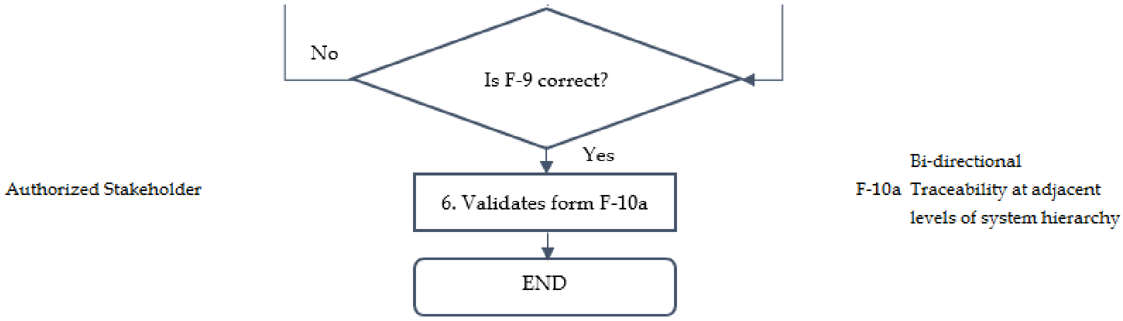
Set of System Requirements: The completeness of SyR
Feasibility: Must be achievable using existent technologies and manufacturing; can be satisfied by a solution that is feasible within life-cycle
Consistency: Does not have individual requirements contradictories
Facilitator: Who leads the collaborative work session
Participants: The stakeholders that participate in the collaborative session.

5. Generalities

Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram





7. Description

1. The Project Manager defines the Facilitator
2. The facilitator integrates all SyR stated in form *F-9 List of System Requirements* in the form *F-10a Traceability at adjacent levels of System Requirements hierarchy*
3. The facilitator establishes a collaborative session and invites all the Stakeholders; during the collaborative session every pair of SyR are analyzed, to identify how, how strong and in what sense (positive or negative) they are co-related
4. The Facilitator fills the left down colored section of form F-10a Traceability at adjacent levels of System Requirements according to the following code:
   ++ Strong positive: direct and strong co-relation between the SyR
   \+ Medium positive: direct and weak co-relation between the SyR
   \- Medium negative: indirect and weak co-relation between the SyR
   -- Strong negative: indirect and strong co-relation between the SyR
   (nothing) If there is no co-relation between the SyR
   Once all relationships are defined, the feasibility and consistency of every pair must be assessed. This way it is assured that the system will be feasible and consistent.
   IMPORTANT NOTE: If available, this paired comparison may be done by specialized software. This way, contradictions can be detected automatically.
5. When a pair of SyR is not feasible nor consistent, the participants must negotiate until making consensus.

Instructions to fill the form:

a. Enter the Win conditions, it is, the combined pair of SyR (quality characteristics) that show a negative co-relation (strong or medium)
b. Enter the Issues that summarize the involved conflict among the Win conditions
c. Restate the initial problem
d. Separate the opposite physical contradictions:
   a. Numerically, considering: a) the different locations of the system, b) the different life stages of the system, and c) the different conditions of the system at different times.
   b. Graphically, considering: a) separation in time, b) separation in space, c) system transformations, and d) phase transformations, or physical-chemical transformation of substances
e. Replace the quality characteristics with general physical characteristics
f. Refers to Altshuller's Contradiction Matrix
g. Derive from among the 40 principles those principles that possibly resolve the system conflict
h. Prepare the candidate options addressing every issue.
i. The Stakeholders evaluate the options to converge on a mutually satisfactory option
j. The Option is formally expressed in the Agreement Schema
k. The facilitator verify the correct content of the form F-10b Stakeholder Negotiation
l. The authorized stakeholder validates it.

6. The facilitator, with aid of the participants, verify the correct content of the form *F-10a Traceability at adjacent levels of System Requirements*. Once the form *F-10a Traceability at adjacent levels of System Requirements* is verified, the authorized stakeholder validate it.

8. Documents

F - 1a	Identify and Weight the Stakeholders
F - 9	List of System Requirements
F - 10a	Bi-directional Traceability at adjacent levels of the system hierarchy
F - 10b	Stakeholder Negotiation "*Win-TRIZ*"

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

W-11 System Requirements

DREAM	**Work Instruction** **System Requirements**	ID Work Instruction W – 11 Version 001

1. Objective

To assign System Requirement (SyR) attributes
To identify the related Stakeholders of each SyR
To identify the Stakeholders responsible for verification and validation (V&V) activities
To verify the quality of the SyR statement
To verify and validate (V&V) objectively the SyR
To identify and mitigate the potential risk that could be generated by SyR
To identify the SyR upward traceability
To identify the SyR bi-directional traceability and its co-relationships

2. Responsible

The Project Manager is responsible for the correct application of this Work Instruction (W)

3. Scope

This W may be applied to New System Development projects

4. Definitions

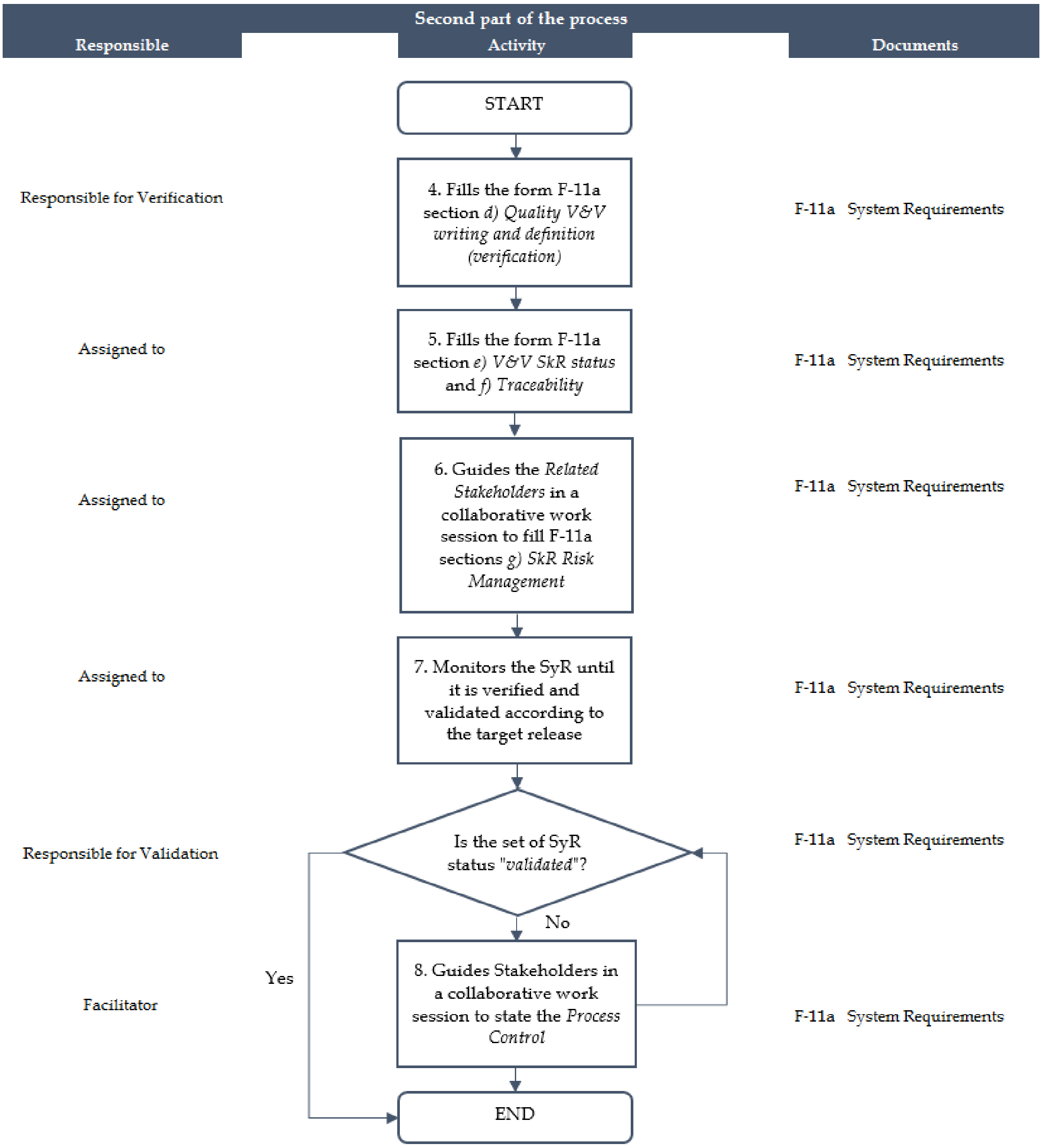
Attribute: additional information included with a requirement statement, which is used to aid in the management of that requirement.
Verify: To verify a stakeholder requirement or a system requirement is to check the application of syntactic and grammatical rules, and characteristics defined in the stakeholder requirements definition process, and the system requirements definition process such as; necessity, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, and verifiable.
Validate: To validate a stakeholder requirement, make sure its content is justified and relevant to stakeholders' expectations, complete and expressed in the language of the customer or end user.
Facilitator: Who leads the collaborative work session.
Participants: The stakeholders that participate in the collaborative session.

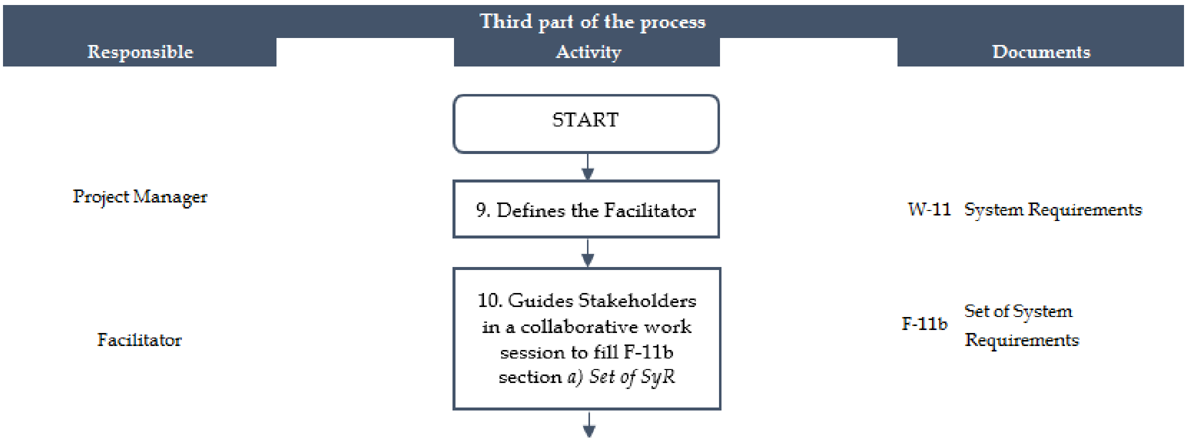
5. Generalities

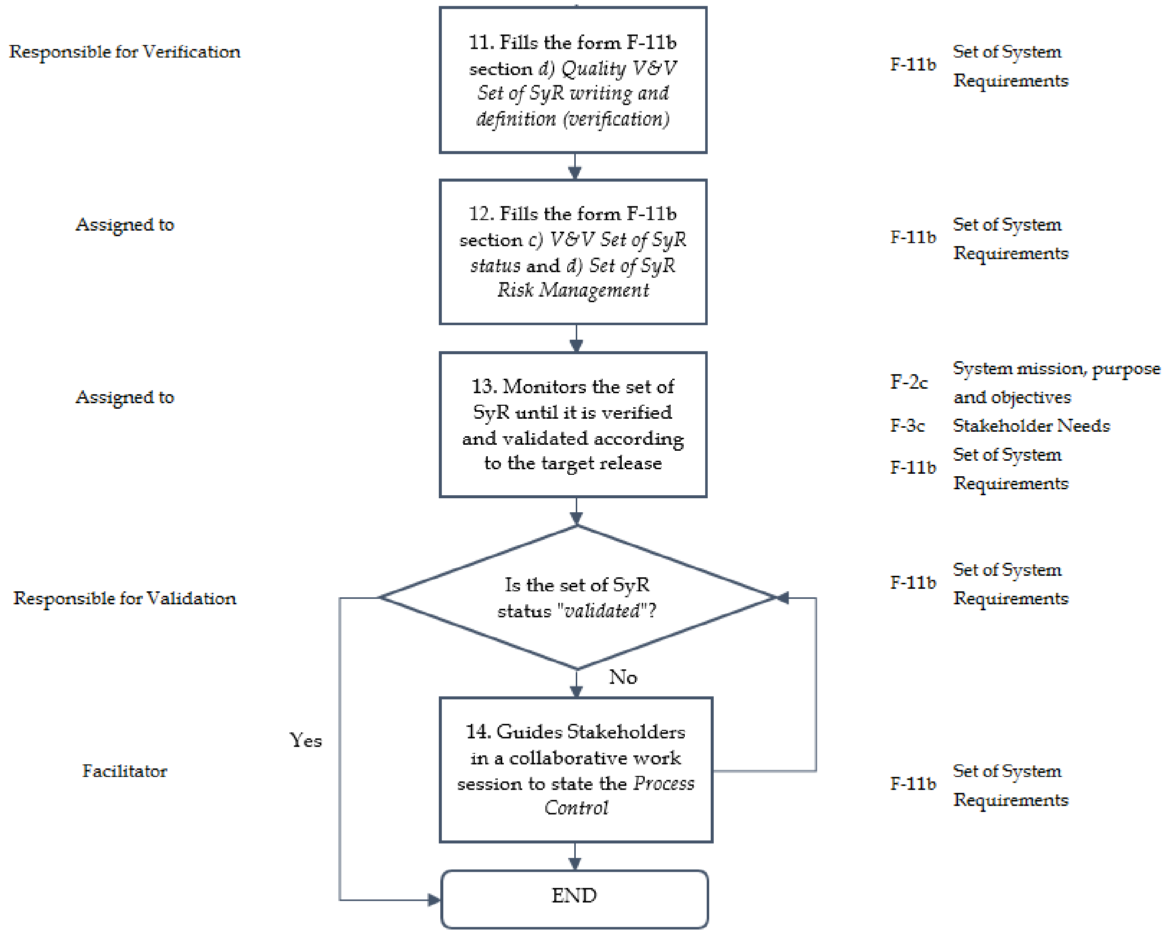
Collaborative work and the application of the Nominal Group Technique is strongly recommended.
Collaborative sessions may be face-to-face or virtual
During the collaborative sessions, it is suggested to have available: chat, audio, and video, in this way the Participants will be able to know how the progress is developing, and their participation will be facilitated.
It is suggested to record the sessions for future reference.

6. Flow diagram

	Fist part of the process	
Responsible	**Activity**	**Documents**
	START	
Project Manager	1. Defines the Facilitator	W-11 System Requirements
Facilitator	2. Generates one record of the form F-11a for each SyR contained in F-9 Fills the section *a) System Requirement (SyR)*	F-9 List of System Requirements F-11a System Requirements
Facilitator	3. Guides Stakeholders in a collaborative work session to fill F-11a sections *b) Related Stakeholders* and *c) SyR Management* for each SyR	F-11a System Requirements
	END	


Second part of the process
Responsible
Activity
Documents
START
Responsible for Verification
4. Fills the form F-11a section *d) Quality V&V writing and definition (verification)*
F-11a System Requirements
Assigned to
5. Fills the form F-11a section *e) V&V SkR status* and *f) Traceability*
F-11a System Requirements
Assigned to
6. Guides the *Related Stakeholders* in a collaborative work session to fill F-11a sections *g) SkR Risk Management*
F-11a System Requirements
Assigned to
7. Monitors the SyR until it is verified and validated according to the target release
F-11a System Requirements
Responsible for Validation
Is the set of SyR status "*validated*"?
F-11a System Requirements
No
Yes
Facilitator
8. Guides Stakeholders in a collaborative work session to state the *Process Control*
F-11a System Requirements
END


Third part of the process
Responsible
Activity
Documents
START
Project Manager
9. Defines the Facilitator
W-11 System Requirements
Facilitator
10. Guides Stakeholders in a collaborative work session to fill F-11b section *a) Set of SyR*
F-11b Set of System Requirements



7. Description

The first part of the process

1. The Project Manager defines the Facilitator
2. The Facilitator generates one record of form *F - 11a System Requirement* for each SyR contained in *F - 9 List of System Requirements*, and fills the section *a) System Requirement (SyR)*
3. The facilitator establishes a collaborative session and invite the Stakeholders; during the collaborative session the Facilitator guides the participants through the analysis of each SyR to fill sections *b) Related Stakeholders* in every stage of the system life cycle (included the Stakeholder authors of the correlated SyNs); and *c) SyR management*: "Assigned to" (stakeholder responsible of this SyR), "Responsible for verification", and "Responsible for validation"

NOTE: To reduce wasting time in the process, it is suggested that the collaborative sessions for the *Second part of the process* should be performed only among the *Related Stakeholders* of each SyR.

The second part of the process

4. The "Responsible for Verification" and the "Responsible for Validation" fill the section *d) Quality: V&V writing and definition*.
5. The "Assigned to" fills the section *e) V&V SkR status* and *f) Traceability*
6. After discussion and consensus among the "Related Stakeholders", the "Assigned to" fills the section *g) SyR Risk Management* according to the following information:

Likelihood		
Score	Likelihood of Occurrence (p)	
5	Near certainty	$p > 80\%$
4	Highly Likely	$60\% < p \leq 80\%$
3	Likely	$40\% < p \leq 60\%$
2	Low likelihood	$20\% < p \leq 40\%$
1	Not likely	$p \leq 20\%$

CONSEQUENCE					
Score	1	2	3	4	5
Performance	Minimal consequence to objectives/goals	Minor consequence to objectives/goals	Unable to achieve a particular objective/ goal, but remaining objective goals represent better than minimum success or outcome	Unable to achieve multiple objectives/goals but minimum success can still be achieved or claimed	Unable to achieve objectives/goals such that minimum success cannot be achieved or claimed
Safety Human	Discomfort or nuisance	First aid event per OSHA criteria	No lost time injury or illness per OSHA criteria	Lost time injury or illness per OSHA criteria	Loss of life
Asset	Minimal consequence: asset has no sign of physical damage	Minor consequence: asset has cosmetic damage and is repairable	Minor consequence: asset is damaged but repairable	Major consequence asset is substantially damaged but repairable	Destroyed: asset is compromised, and un-repairable: a total loss
Schedule	Minimal consequence	Critical path is not slipped; total slack of slipped tasks will not impact critical path in less than 10 days	Critical path is not slipped; total slack of slipped tasks is within 10 days of impacting the critical path	Critical path slips	Critical path slips and one or more critical milestones or events cannot be met
Cost	Minimal consequence	Minor cost consequence. Cost variance ≤ 5% of total approved FY baseline	Cost consequence. Cost variance >5% but ≤ of total approved FY baseline	Cost consequence. Cost variance >10% but <15% of total approved FY baseline	Major cost consequence. Cost consequence. Cost variance >15% of total approved FY baseline

7. The "Assigned to" monitors the SyR until it is verified and validated according to the target release, and its status is "accepted"
8. The Facilitator guides the Stakeholders in a collaborative work session to state the *Process Control*, corrective actions due to non-conformities detection

The third part of the process

9. The Project Manager defines the Facilitator
10. The facilitator establishes a collaborative session and invites the Stakeholders; during the collaborative session the Facilitator guides the Stakeholders to fill the form *F-11b Set of System Requirements* section *a) Set of SyR Management*
11. The "Responsible for Verification" fills the form *F-11b Set of System Requirements* section *d) Quality V&V Set of SyR writing and definition (verification)*
12. The "Assigned to" fills the form *F-11b Set of System Requirements* section *c) V&V Set of SyR status* and *d) Set of SyR Risk Management*
13. The "Assigned to" monitors the set of SkR until it is verified and validated according to the target release, and its status is "accepted"
    To validate the set of SyR:
    a. If a prototype or model is built: ***Does the prototype or model meet the stakeholder needs?***
    b. Are the records contained in the forms *F-2c System mission, purpose, and objectives* and *F-3c Stakeholder needs* congruent with the set of SyR?
    c. Analyze the comparison between the results obtained versus the expected ones
    d. In case of detecting a non-conformity, continue with step 14.
14. The Facilitator guides the Stakeholders in a collaborative work session to state the *Process Control*, corrective actions due to non-conformities detection

8. Documents

F - 2c	System mission, purpose and objectives
F - 3c	Stakeholder needs
F - 9	List of System Requirements
F - 11a	System Requirement
F - 11b	Set of System Requirements

9. Work Instruction History

Date	Author	Changes	Authorized by	Version
29/03/2019	K GOMEZ	Thesis version		001

DREAM Forms

F-1a Identify and Weight the Stakeholders (extract)

DREAM	Form **Identify and Weight the Stakeholders**	ID Form **F - 1a** Version **006**

System:		**Participants:**	
Date:			
Facilitator:		**Validated by:**	

	Life Cycle Stages and their purposes	Concept Identify Stakeholders needs Evaluate alternate concepts Recommend possible solutions													
	Stakeholders types	Stakeholders	Power				Legitimacy				Urgency				T
			0	1	2	3	0,0	0,5	1,0	1,5	##	0,25	0,50	0,75	
Professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for operation that will affect the development and/or operation of the system	**1. Legislators**														0
															0
															0
															0
															0
															0
Who pay, by, pay for its operation, and obtain revenue	**2. Owner or investor**														0
															0
															0
															0
															0
Who prior to the Project, identify the need for a new asset and the potential benefit it will bring	**3. Project executive or project sponsor**														0
															0
															0
															0
															0
Who buy the product the new assets produces	**4. Consumers**														0
															0
															0
															0
															0
Who operate the asset on behalf of the owner	**5. Operators/users**														0
															0
															0
															0
															0
Who manage the project, the analysis and design team in the requirement engineering process	**6. Project manager and Project team**														0
															0
															0
															0
															0
Senior management in the lead contractor	**7. Senior supplier (design and/or management)**														0
															0
															0
															0
															0
People or groups who provides goods, materials; Works or services	**8. Other suppliers**														0
															0
															0
															0
The public concerned with environmental and social impacts of the system, they will want to know how their taxes have been spent	**9. Public**														0
															0
															0
															0
Persons who don't want the system or malevolant	**10. People who don't want the system or maliciuosly intended**														0
															0
															0
															0
															0
															0

F-1b List of Stakeholders (extract)

DREAM

Form
List of Stakeholders

ID Form: **F - 1b**
Version: **003**

System:
Date:
Facilitator:

Participants:
Validated by:

	Life Cycle Stages and their purposes	Concept Identify Stakeholders needs Evaluate alternate concepts Recommend possible solutions									
	Stakeholders types	**Stakeholders**	**Stk ID**	**Company name**	**Specific role**	**Readiness**	**Domain**	**Expertise level**	**Objective / Interest in the project**	**e-mail**	**Telephone**
Professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for operation that will affect the development and/or operation of the system	**1. Legislators**										
Who pay, by, pay for its operation, and obtain revenue	**2. Owner or investor**										
Who prior to the Project, identify the need for a new asset and the potential benefit it will bring	**3. Project executive or project sponsor**										
Who buy the product the new assets produces	**4. Consumers**										

F-2a Define system mission, purpose, and objectives

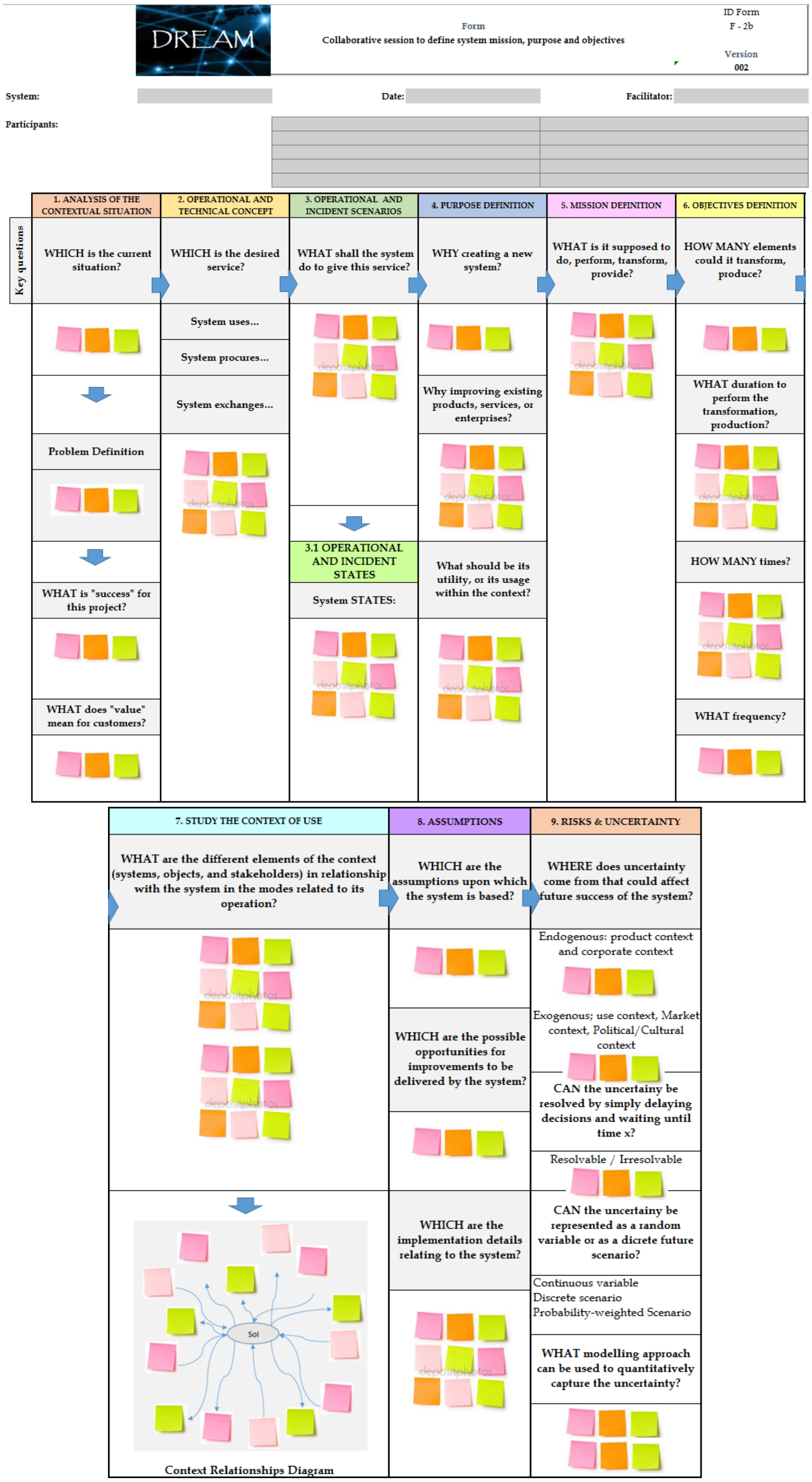
DREAM	Form Define system mission, purpose and objectives	ID Form F - 2a Version 002

System: | Date: | Stakeholder:

	1. ANALYSIS OF THE CONTEXTUAL SITUATION	2. OPERATIONAL AND TECHNICAL CONCEPT	3. OPERATIONAL AND INCIDENT SCENARIOS	4. PURPOSE DEFINITION	5. MISSION DEFINITION	6. OBJECTIVES DEFINITION
Key questions	WHICH is the current situation?	WHICH is the desired service?	WHAT shall the system do to give this service?	WHY creating a new system?	WHAT is it supposed to do, perform, transform, provide?	HOW MANY elements could it transform, produce?
		System uses...				
		System procures...				
		System exchanges...		Why improving existing products, services, or enterprises?		WHAT duration to perform the transformation, production?
	Problem Definition					
	WHAT is "success" for this project?		3.1 OPERATIONAL AND INCIDENT STATES System STATES:	What should be its utility, or its usage within the context?		HOW MANY times?
	WHAT does "value" mean for customers?					WHAT frequency?

7. STUDY THE CONTEXT OF USE	8. ASSUMPTIONS	9. RISKS & UNCERTAINTY
WHAT are the different elements of the context (systems, objects, and stakeholders) in relationship with the system in the modes related to its operation?	WHICH are the assumptions upon which the system is based?	WHERE does uncertainty come from that could affect future success of the system?
		Endogenous: product context and corporate context
		Exogenous; use context, Market context, Political/Cultural context
	WHICH are the possible opportunities for improvements to be delivered by the system?	CAN the uncertainty be resolved by simply delaying decisions and waiting until time x?
		Resolvable / Irresolvable
	WHICH are the implementation details relating to the system?	CAN the uncertainty be represented as a random variable or as a dicrete future scenario?
		Continuous variable Discrete scenario Probability-weighted Scenario
SoI		WHAT modelling approach can be used to quantitatively capture the uncertainty?
Context Relationships Diagram		

F-2b Collaborative session to define system mission, purpose, and objectives



F-2c System mission, purpose and objectives

DREAM	Form System mission, purpose and objectives	ID Form F - 2c Version 003

System: | **Date:** | **Facilitator:**

Participants:

Validated by:

	1. ANALYSIS OF THE CONTEXTUAL SITUATION	2. OPERATIONAL AND TECHNICAL CONCEPT	3. OPERATIONAL AND INCIDENT SCENARIOS	4. PURPOSE DEFINITION	5. MISSION DEFINITION	6. OBJECTIVES DEFINITION
Key questions	WHICH is the current situation?	WHICH is the desired service?	WHAT shall the system do to give this service?	WHY creating a new system?	WHAT is it supposed to do, perform, transform, provide?	HOW MANY elements could it transform, produce?
		System uses...				
		System procures...				
		System exchanges...		Why improving existing products, services, or enterprises?		WHAT duration to perform the transformation, production?
	Problem Definition					
			3.1 OPERATIONAL AND INCIDENT STATES	What should be its utility, or its usage within the context?		HOW MANY times?
	WHAT is "success" for this project?		System STATES:			
	WHAT does "value" mean for customers?					WHAT frequency?

7. STUDY THE CONTEXT OF USE	8. ASSUMPTIONS	9. RISKS & UNCERTAINTY
WHAT are the different elements of the context (systems, objects, and stakeholders) in relationship with the system in the modes related to its operation?	WHICH are the assumptions upon which the system is based?	WHERE does uncertainty come from that could affect future success of the system?
		Endogenous: product context and corporate context
	WHICH are the possible opportunities for improvements to be delivered by the system?	Exogenous; use context, Market context, Political/Cultural context
		CAN the uncertainty be resolved by simply delaying decisions and waiting until time x?
		Resolvable / Irresolvable
	WHICH are the implementation details relating to the system?	CAN the uncertainty be represented as a random variable or as a dicrete future scenario?
		Continuous variable Discrete scenario Probability-weighted Scenario
Sol		WHAT modelling approach can be used to quantitatively capture the uncertainty?
Context Relationships Diagram		

F-3a Elicit Stakeholder Needs (extract)

DREAM	Form **Elicit Stakeholder Needs**	ID Form **F - 3a** Version **004**

System: **Date:** **Stakeholder:**

	QUESTION	ID	NEED	RATIONAL
1. Functional Requirements	What are the main operational activities or highest level functions that the system of interest has to achieve?	SkN _ 1.0.0 _ 1		
		SkN _ 1.0.0 _ 2		
		SkN _ 1.0.0 _ 3		
		SkN _ 1.0.0 _ 4		
		SkN _ 1.0.0 _ 5		
2. Effectiveness / Performance Requirements	What is the expected performance or effectiveness of the system? What is the quality metric of effectiveness? Describe the measure of effectiveness (performances) expected by the system to satisfy the expected service (quantitative data as much as possible or at least qualitative data).	SkN _ 2.0.0 _ 1		
		SkN _ 2.0.0 _ 2		
		SkN _ 2.0.0 _ 3		
		SkN _ 2.0.0 _ 4		
		SkN _ 2.0.0 _ 5		
		SkN _ 2.0.0 _ 6		
		SkN _ 2.0.0 _ 7		
		SkN _ 2.0.0 _ 8		
3. Interface Requirements	What are the functional interfaces between the system and the components of its operational context? Note: the human-system interfaces are described in the section of ergonomics	SkN _ 3.1.0 _ 1		
		SkN _ 3.1.0 _ 2		
		SkN _ 3.1.0 _ 3		
		SkN _ 3.1.0 _ 4		
		SkN _ 3.1.0 _ 5		
	What are the physical interfaces that connect the system to the components of its context of use? Describe. These physical interfaces may be: electrical cables, connectors, pipes, data format, protocols, etc.	SkN _ 3.2.0 _ 6		
		SkN _ 3.2.0 _ 7		
		SkN _ 3.2.0 _ 8		
		SkN _ 3.2.0 _ 9		
		SkN _ 3.2.0 _ 10		
	Which are the Operational Modes (on/off, standby, run, maintenance, etc.)? What is the system expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one?	SkN _ 4.1.1 _ 1		
		SkN _ 4.1.1 _ 2		
		SkN _ 4.1.1 _ 3		
		SkN _ 4.1.1 _ 4		
		SkN _ 4.1.1 _ 5		
	Describe each Operational Scenario (sequence, concurrence of actions) of the system and the exchanges with the components of the context, including the actions of the users.	SkN _ 4.1.2 _ 1		
		SkN _ 4.1.2 _ 2		
		SkN _ 4.1.2 _ 3		
		SkN _ 4.1.2 _ 4		
		SkN _ 4.1.2 _ 5		
	What are the Incident Modes? What the system is expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one?	SkN _ 4.1.3 _ 1		
		SkN _ 4.1.3 _ 2		
		SkN _ 4.1.3 _ 3		
		SkN _ 4.1.3 _ 4		
		SkN _ 4.1.3 _ 5		

F-3b Collaborative Session Elicit Stakeholder Needs

DREAM	Form Collaborative session to elicit Stakeholder Needs	ID Form F - 3b Version 004

System: **Date:** **Facilitator:**

Participants:

Key questions	Modes and Scenarios	1. Expected Services	2. Expected Effectiveness or Performance	3. Interfaces	4. Operational Conditions
	WHAT are the Operational and Incident Modes? WHAT the system is expected to do in each mode?	WHAT are the desired services? WHAT are the main operational activities or functions to achieve?	WHAT is the expected performance or effectiveness? WHAT is the quality metric of effectiveness?	WHAT are the Functional interfaces between the system and the components of its context of use?	WHAT are the environmental conditions to which the system is submitted? Examples: temperature, pressure, vibrations, acoustic noise, geographic environment, …
	WHAT are de Operational and Incident Scenarios? What are the trigger events that initiate the transition from one mode to another one?			WHAT are de Physical interfaces that connect the system to the components of its context of use?	WHAT are the resources consumed or produced by the system which de not belong to the system?
					WHAT are the requirements concern to Transportation, Storage, and handling conditions?
					WHAT are the Ergonomics and Human Factors? WHICH are the expected man-system interfaces? Examples: keyboard, buttons, screen, …
					HOW does the system provide confidence to users in the service that it returns to them?
					WHAT is the documented information concerns installation, operation and utilization of the system? Preventive maintenance procedures? Trainings? …
					Safety: Availability: Survivability: Reliability: Mantainability: Modularity: Interoperability: Reconfigurability: Modifiability: Extensibility: Scalability: Flexibility: Adaptability: Versatility: Evolvability: Changeability Information Availability: Information Integrity: Information Confidentialy:
					WHAT are the Maintenance or Logistic Support conditions? Examples: duration of maintenance, qualification of the maintenance team, …

5. Constraints		6. Forensic Requirements	7. Validation Requirements
WHAT are the physical constraints such size, weight, colour, volume?		WHAT are the requirements for conducting a forensic investigation in order to establish the cause of the incident and how it can be prevented in the future?	WHAT are the validation Requirements? WHICH are the elements or arguments to select the most effective solution among candidate solutions?
WHAT are the implementation constraints? WHAT are the imposed solutions?	WHAT are the extension constraints for future evolutions?		WHO is authorized to validate the system?
WHAT are the transfer for use constraints? Examples: acces to site, available means of lifting, … WHAT are the tasks and organization to install the system?	WHAT are the disposal constraints? Destruction or Storage?		

F-3c Stakeholder Needs (extract)

	Form	ID Form
DREAM	Form **Elicit Stakeholder Needs**	**F - 3c**
		Version **005**

System: ___ **Date:** ___ **Facilitator:** ___

Participants:

Validated by: ___

	QUESTION	ID	STAKEHOLDER NEED (SkN)	RATIONAL	AUTHOR
1. Functional Requirements	What are the main operational activities or highest level functions that the system of interest has to achieve?	SkN_1.0.0_1			
		SkN_1.0.0_2			
		SkN_1.0.0_3			
		SkN_1.0.0_4			
		SkN_1.0.0_5			
2. Effectiveness / Performance Requirements	What is the expected performance or effectiveness of the system? What is the quality metric of effectiveness? Describe the measure of effectiveness (performances) expected by the system to satisfy the expected service (quantitative data as much as possible or at least qualitative data).	SkN_2.0.0_1			
		SkN_2.0.0_2			
		SkN_2.0.0_3			
		SkN_2.0.0_4			
		SkN_2.0.0_5			
		SkN_2.0.0_6			
		SkN_2.0.0_7			
		SkN_2.0.0_8			
3. Interface Requirements	What are the functional interfaces between the system and the components of its operational context? Note: the human-system interfaces are described in the section of ergonomics	SkN_3.1.0_1			
		SkN_3.1.0_2			
		SkN_3.1.0_3			
		SkN_3.1.0_4			
		SkN_3.1.0_5			
	What are the physical interfaces that connect the system to the components of its context of use? Describe. These physical interfaces may be: electrical cables, connectors, pipes, data format, protocols, etc.	SkN_3.2.0_6			
		SkN_3.2.0_7			
		SkN_3.2.0_8			
		SkN_3.2.0_9			
		SkN_3.2.0_10			
	Which are the Operational Modes (on/off, standby, run, maintenance, etc.)? What is the system expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one?	SkN_4.1.1_1			
		SkN_4.1.1_2			
		SkN_4.1.1_3			
		SkN_4.1.1_4			
		SkN_4.1.1_5			
	Describe each Operational Scenario (sequence, concurrence of actions) of the system and the exchanges with the components of the context, including the actions of the users.	SkN_4.1.2_1			
		SkN_4.1.2_2			
		SkN_4.1.2_3			
		SkN_4.1.2_4			
		SkN_4.1.2_5			
	What are the Incident Modes? What the system is expected to do in each mode? What are the trigger events that initiate the transition from one mode to another one?	SkN_4.1.3_1			
		SkN_4.1.3_2			
		SkN_4.1.3_3			
		SkN_4.1.3_4			
		SkN_4.1.3_5			

F-4a Stakeholder Needs feasibility and consistency (extract)

DREAM	Form **Stakeholder Needs feasibility and consistency**	ID Form **F - 4a** Version **006**

System: **Date:** **Facilitator:**

Participants:

Validated by:

		Status: active A / cancelled C	1. Functional Requirements					2. Effectiveness / Performance Requirements								3. Interface Requirements									
			SkN_1.0.0_1	SkN_1.0.0_2	SkN_1.0.0_3	SkN_1.0.0_4	SkN_1.0.0_5	SkN_2.0.0_1	SkN_2.0.0_2	SkN_2.0.0_3	SkN_2.0.0_4	SkN_2.0.0_5	SkN_2.0.0_6	SkN_2.0.0_7	SkN_2.0.0_8	SkN_3.1.0_1	SkN_3.1.0_2	SkN_3.1.0_3	SkN_3.1.0_4	SkN_3.1.0_5	SkN_3.2.0_6	SkN_3.2.0_7	SkN_3.2.0_8	SkN_3.2.0_9	SkN_3.2.0_10
1. Functional Requirements	SkN_1.0.0_1																								
	SkN_1.0.0_2																								
	SkN_1.0.0_3																								
	SkN_1.0.0_4																								
	SkN_1.0.0_5																								
2. Effectiveness / Performance Requirements	SkN_2.0.0_1																								
	SkN_2.0.0_2																								
	SkN_2.0.0_3																								
	SkN_2.0.0_4																								
	SkN_2.0.0_5																								
	SkN_2.0.0_6																								
	SkN_2.0.0_7																								
	SkN_2.0.0_8																								
3. Interface Requirements	SkN_3.1.0_1																								
	SkN_3.1.0_2																								
	SkN_3.1.0_3																								
	SkN_3.1.0_4																								
	SkN_3.1.0_5																								
	SkN_3.2.0_6																								
	SkN_3.2.0_7																								
	SkN_3.2.0_8																								
	SkN_3.2.0_9																								
	SkN_3.2.0_10																								

F-4b Stakeholder Negotiation

DREAM	Form **Stakeholder Negotiation**	ID Form **F - 4b** Version **002**

System:

Date:

Facilitator:

Participants:

Validated by:

WinWin negotiation model

1. Win Conditions	**2. Issues**
SkN with Negative (medium or strong) co-relationship	Summarize the involved conflict among the Win conditions
4. Agreements	**3. Options**
Mutually satisfactory option	Candidate options are prepared and evaluated

F-5 Prioritize Stakeholder Needs (extract)

DREAM	Form **Prioritize Stakeholder Needs**	ID Form **F - 5** Version **006**

System: **Date:** **Facilitator:**

Participants:

Validated by:

MATRIX A		1. Functional Requirements					2. Effectiveness / Performance Requirements								3. Interface Requirements									
		SkN _ 1.0.0 _ 1	SkN _ 1.0.0 _ 2	SkN _ 1.0.0 _ 3	SkN _ 1.0.0 _ 4	SkN _ 1.0.0 _ 5	SkN _ 2.0.0 _ 1	SkN _ 2.0.0 _ 2	SkN _ 2.0.0 _ 3	SkN _ 2.0.0 _ 4	SkN _ 2.0.0 _ 5	SkN _ 2.0.0 _ 6	SkN _ 2.0.0 _ 7	SkN _ 2.0.0 _ 8	SkN _ 3.1.0 _ 1	SkN _ 3.1.0 _ 2	SkN _ 3.1.0 _ 3	SkN _ 3.1.0 _ 4	SkN _ 3.1.0 _ 5	SkN _ 3.2.0 _ 6	SkN _ 3.2.0 _ 7	SkN _ 3.2.0 _ 8	SkN _ 3.2.0 _ 9	SkN _ 3.2.0 _ 10
1. Functional Requirements	SkN _ 1.0.0 _ 1	1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 1.0.0 _ 2		1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 1.0.0 _ 3			1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 1.0.0 _ 4				1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 1.0.0 _ 5					1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
2. Effectiveness / Performance Requirements	SkN _ 2.0.0 _ 1						1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 2							1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 3								1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 4									1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 5										1,00	####	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 6											1,00	####	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 7												1,00	####	####	####	####	####	####	####	####	####	####	####
	SkN _ 2.0.0 _ 8													1,00	####	####	####	####	####	####	####	####	####	####
3. Interface Requirements	SkN _ 3.1.0 _ 1														1,00	####	####	####	####	####	####	####	####	####
	SkN _ 3.1.0 _ 2															1,00	####	####	####	####	####	####	####	####
	SkN _ 3.1.0 _ 3																1,00	####	####	####	####	####	####	####
	SkN _ 3.1.0 _ 4																	1,00	####	####	####	####	####	####
	SkN _ 3.1.0 _ 5																		1,00	####	####	####	####	####
	SkN _ 3.2.0 _ 6																			1,00	####	####	####	####
	SkN _ 3.2.0 _ 7																				1,00	####	####	####
	SkN _ 3.2.0 _ 8																					1,00	####	####
	SkN _ 3.2.0 _ 9																						1,00	####
	SkN _ 3.2.0 _ 10																							1,00
	TOTAL	1,00	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####	####

Consistency Ratio:

$$CR = \frac{\lambda_{max} - N}{(N-1)RI}$$

Where: λ_{max} = maximal eigenvalue
N = dimension of the matrix
RI = random index, see Table 32

Consistency Ratio

F-6 List of Stakeholder Requirements (extract)

DREAM

Form
List of Stakeholder Requirements

ID Form
F - 6

Version
005

Participants:

System: **Date:** **Facilitator:**

Validated by:

	ID	STAKEHOLDER NEED (SkN)	ID	STAKEHOLDER REQUIREMENT (SkR)	ID	DERIVED OR DECOMPOSED STAKEHOLDER REQUIREMENT (SkR) (if needed)
1.Functional Requirements	SkN_1.0.0_1		SkR_1.0.0_1		SkR_1.0.0_1_1	
	SkN_1.0.0_2		SkR_1.0.0_2		SkR_1.0.0_2_1	
	SkN_1.0.0_3		SkR_1.0.0_3		SkR_1.0.0_3_1	
	SkN_1.0.0_4		SkR_1.0.0_4		SkR_1.0.0_4_1	
	SkN_1.0.0_5		SkR_1.0.0_5		SkR_1.0.0_5_1	
2. Effectiveness I Performance Requirements	SkN_2.0.0_1		SkR_2.0.0_1		SkR_2.0.0_1_1	
	SkN_2.0.0_2		SkR_2.0.0_2		SkR_2.0.0_2_1	
	SkN_2.0.0_3		SkR_2.0.0_3		SkR_2.0.0_3_1	
	SkN_2.0.0_4		SkR_2.0.0_4		SkR_2.0.0_4_1	
	SkN_2.0.0_5		SkR_2.0.0_5		SkR_2.0.0_5_1	
	SkN_2.0.0_6		SkR_2.0.0_6		SkR_2.0.0_6_1	
	SkN_2.0.0_7		SkR_2.0.0_7		SkR_2.0.0_7_1	
	SkN_2.0.0_8		SkR_2.0.0_8		SkR_2.0.0_8_1	
3. Interface Requirements	SkN_3.1.0_1		SkR_3.1.0_1		SkR_3.1.0_1_1	
	SkN_3.1.0_2		SkR_3.1.0_2		SkR_3.1.0_2_1	
	SkN_3.1.0_3		SkR_3.1.0_3		SkR_3.1.0_3_1	
	SkN_3.1.0_4		SkR_3.1.0_4		SkR_3.1.0_4_1	
	SkN_3.1.0_5		SkR_3.1.0_5		SkR_3.1.0_5_1	
	SkN_3.2.0_6		SkR_3.2.0_6		SkR_3.2.0_6_1	
	SkN_3.2.0_7		SkR_3.2.0_7		SkR_3.2.0_7_1	
	SkN_3.2.0_8		SkR_3.2.0_8		SkR_3.2.0_8_1	
	SkN_3.2.0_9		SkR_3.2.0_9		SkR_3.2.0_9_1	
	SkN_3.2.0_10		SkR_3.2.0_10		SkR_3.2.0_10_1	

F-7a Stakeholder Requirements

DREAM	Form **Stakeholder Requirement**	ID Form **F - 7a** Version **004**

a) STAKEHOLDER REQUIREMENT (SkR)					
ID	SkR	Value	Rational: Why does this need and expectation exist?	Author	
				ID	Stakeholder
If available paste here the drawing, table, file, site, etc. related to SkR					

b) RELATED STAKEHOLDERS

ID	Stakeholder	ID	Stakeholder	ID	Stakeholder	ID	Stakeholder	ID	Stakeholder

c) SkR MANAGEMENT		
Stakeholder Role	ID	Stakeholder
Assigned to		
Responsible for verification		
Responsible for validation		

d) QUALITY V&V SkR writing and definition

	Is the requirement…?	Yes	No
Verification	**Mature:** Is the expression of the SkR close to Stakeholders expectatives?	✓	×
	Accurate: Did the Stakeholder express their expectatives with precision?	✓	×
	Feasible: Was the SkR feasibility assessed through identified operational concepts?	✓	×
	Appropriate to level: Is the SkR appropriate to the level at it is stated?	✓	×
	Complete: Is the SkR explained enough?	✓	×
	Conforming: Is the SkR conform to a standard formal structure?	✓	×
	Verifiable: Can be verified that the system meets or possesses the SkR?	✓	×
	Necessary: Should the system be able to function in the desired way with this SkR?	✓	×
	Singular: Is not the SkR a combination of two or more requirements?	✓	×
	Correct: Is this a SKR that will result in the desired system performance?	✓	×
	Unambiguous: Is there only one interpretation of the SkR?	✓	×
	Implementation free: Does the SkR stament states what is required instead of saying how?	✓	×
	Consistent: Is free of conflicts with other requirements?	✓	×
Validation	**Understanding:** Is this SkR understandable?	✓	×
	Relevance: Do the expression of the requirement allows to define the importance of the solution?	✓	×

Note: If any answer is "No", rewrite the SkR statement until all answers are "Yes"

e) V&V Set of SkR status

Target release dd/mm/yyyy

Verification Method

Verification Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status ✓ × **Approval date** dd/mm/yyyy **Verification Results**

Validation Method

Validation Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status ✓ × **Approval date** dd/mm/yyyy **Validation Results**

Process Control

Corrective Action #	Corrective action	ID	Stakeholder responsible of corrective action

f) TRACEABILITY

Upward traceability: Parent Stakeholder Need (SkN)

ID	SkN	Value	Rational	Author	
				ID	Stakeholder

Bi-directional traceability: Related SkR

ID	Co-relationship (++ + - --)	SkR	Value	Rational	Author	
					ID	Stakeholder
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					

g) SkR RISK MANAGEMENT

Risk Identification

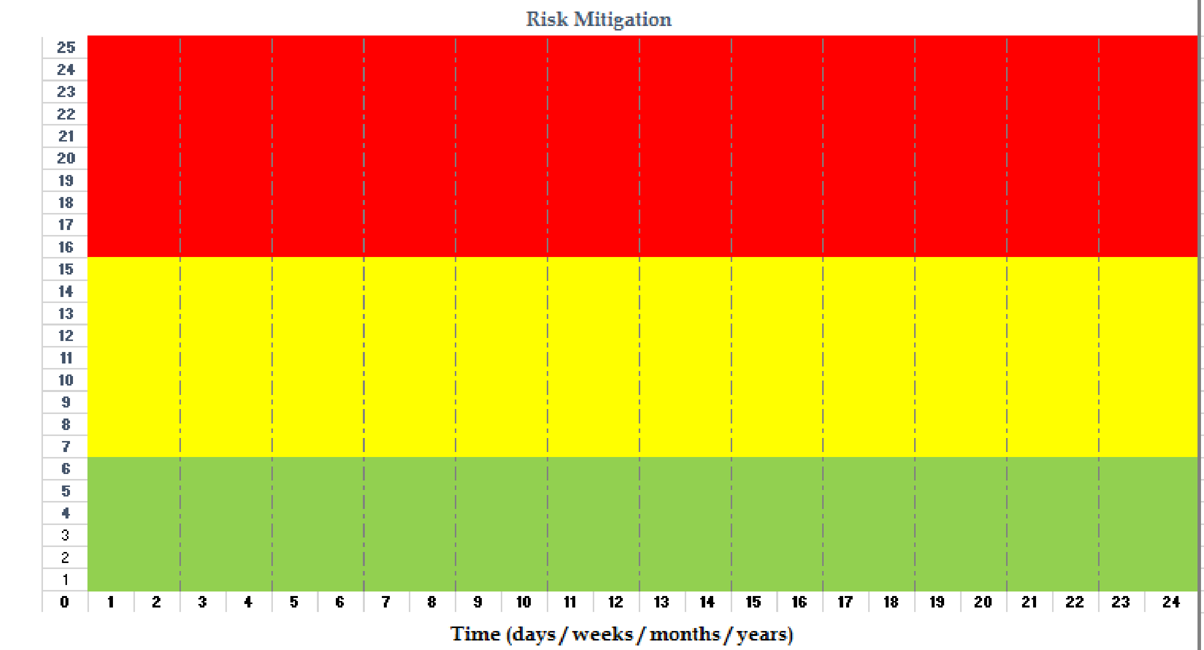
Risk Title	
Risk Statement	
Context Statement	
Closure Criteria	

Likelihood					
5	7	16	20	23	25
4	6	13	18	22	14
3	4	10	15	19	21
2	2	8	11	14	17
1	1	3	5	9	12
	1	2	3	4	5

Consequence

Sunrise	dd/mm/yyyy
Sunset	dd/mm/yyyy
Impact horizont	near/mid/long

Consequence	Rational	Likelihood	Rational



	Key Mitigation Action Planned	Success Criteria	IMS No.	Cost	Start	Completion
01					dd/mm/yyyy	dd/mm/yyyy
02					dd/mm/yyyy	dd/mm/yyyy
03					dd/mm/yyyy	dd/mm/yyyy
04					dd/mm/yyyy	dd/mm/yyyy
05					dd/mm/yyyy	dd/mm/yyyy
06					dd/mm/yyyy	dd/mm/yyyy

F-7b Set of Stakeholder Requirements

DREAM	Form **Set of Stakeholder Requirements**	ID Form **F - 7b** Version **004**

System: **Date:**   **Facilitator:**

Participants:

a) Set of SkR MANAGEMENT		
Stakeholder Role	**ID**	**Stakeholder**
Assigned to		
Responsible for verification		
Responsible for validation		

b) QUALITY V&V Set of SkR writing and definition

	Is the requirement…?	Yes	No
Verification	**Exhaustive:** Were all the Stakeholders identified and interviewed? Were all the Stakeholder Requirements expressed and writen?	✓	×
	Complete: Is the set of SkR explained enough (no need of further amplification)?	✓	×
	Consistent: Is the set of SkR free of contradictory individual SkR ?	✓	×
	Affordable/Feasible: Can the set of SkR be satisfied by a solution that is feasible within life-cycle constraints?	✓	×
	Bounded: Does the set of SkR maintain the identified scope for the intended solution without increasing beyond what is needed?	✓	×
	Comprehensible: Are the set of SkR written such that it is clear as: a) to what is expected by the entity, and b) its relation to the system of which it is a part?	✓	×
	Able to be validated: Is possible to prove that the set of SkR will lead to the satisfaction of needs while respecting the constraints?	✓	×
	Non-redundant: Does NOT the set of SkR have repeated individual SkR?	✓	×
Validation	**Relevance:** the expression of the requirements allows to define the importance of the solution?	✓	×

Note: If any answer is "No", review the set of SkR until all answers are "Yes"

c) V&V Set of SkR status

Target release dd/mm/yyyy

Verification Method

Verification Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status ✓ ×   **Approval date** dd/mm/yyyy   **Verification Results**

Validation Method

Validation Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status ✓ × **Approval date** dd/mm/yyyy **Validation Results**

Process Control			
Corrective Action #	Corrective action	ID	Stakeholder responsible of corrective action

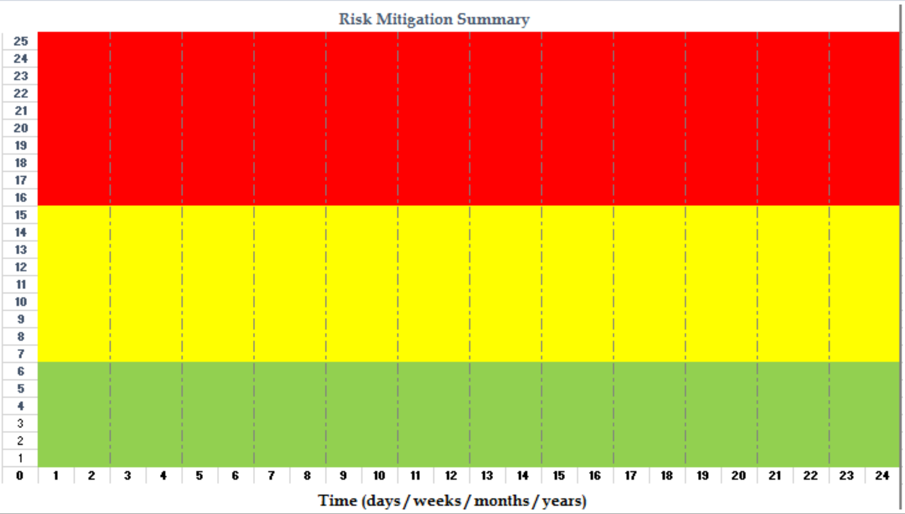
d) Set of SkR RISK MANAGEMENT

Risk Identification Summary

Risk #	Risk Title

Likelihood					
5	7	16	20	23	25
4	6	13	18	22	14
3	4	10	15	19	21
2	2	8	11	14	17
1	1	3	5	9	12
	1	2	3	4	5
	Consequence				

External Risk #	Comments



External Risk #	Risk Title	Start	Completion
		dd/mm/yyyy	dd/mm/yyyy
		dd/mm/yyyy	dd/mm/yyyy
		dd/mm/yyyy	dd/mm/yyyy

F-8 Analyze Stakeholder Requirements

DREAM	Form **Analyze Stakeholders Requirements**	ID Form **F - 8** Version **001**

System: **Date:** **Facilitator:**

Participants:

Validated by:

QUESTIONS

	Yes	No	If "*No*" go to forms:
Are the stakeholders through the system life cycle being identified?	✓	×	F-1a and F-1b
Are the stakeholders through the system life cycle being involved?	✓	×	
Are the assumptions being expressed and understood?	✓	×	F-2c
Has been identified where the value is?	✓	×	
Has been identified what success is for the project?	✓	×	
Have the different scenarios being identified?	✓	×	F-2c F-3c F-4a F-4b F-5 F-6 F-7b
Have been identified the functional and physical interfaces of the system?	✓	×	
Have been identified the flows of material, energy and/or information?	✓	×	
Is all the retrieved information complete and coherent?	✓	×	

F-9 List of System Requirements (extract)

DREAM	Form **List of System Requirements**	ID Form **F - 9** Version **002**

System: **Date:** **Facilitator:**

Participants:

Validated by:

ID	STAKEHOLDER REQUIREMENT (SkR)	ID	DERIVED OR DECOMPOSED STAKEHOLDER REQUIREMENT (SkR) (if needed)
SkR _ 1.0.0 _ 1		SkR _ 1.0.0 _ 1 _ 1	
SkR _ 1.0.0 _ 2		SkR _ 1.0.0 _ 2 _ 1	
SkR _ 1.0.0 _ 3		SkR _ 1.0.0 _ 3 _ 1	
SkR _ 1.0.0 _ 4		SkR _ 1.0.0 _ 4 _ 1	
SkR _ 1.0.0 _ 5		SkR _ 1.0.0 _ 5 _ 1	
SkR _ 2.0.0 _ 1		SkR _ 2.0.0 _ 1 _ 1	
SkR _ 2.0.0 _ 2		SkR _ 2.0.0 _ 2 _ 1	
SkR _ 2.0.0 _ 3		SkR _ 2.0.0 _ 3 _ 1	
SkR _ 2.0.0 _ 4		SkR _ 2.0.0 _ 4 _ 1	
SkR _ 2.0.0 _ 5		SkR _ 2.0.0 _ 5 _ 1	
SkR _ 2.0.0 _ 6		SkR _ 2.0.0 _ 6 _ 1	
SkR _ 2.0.0 _ 7		SkR _ 2.0.0 _ 7 _ 1	
SkR _ 2.0.0 _ 8		SkR _ 2.0.0 _ 8 _ 1	

ID	SYSTEM REQUIREMENT (SyR)	ID	DERIVED OR DECOMPOSED SYSTEM REQUIREMENT (SkR) (if needed)
SyR _ 1.0.0 _ 1		SyR _ 1.0.0 _ 1 _ 1	
SyR _ 1.0.0 _ 2		SyR _ 1.0.0 _ 2 _ 1	
SyR _ 1.0.0 _ 3		SyR _ 1.0.0 _ 3 _ 1	
SyR _ 1.0.0 _ 4		SyR _ 1.0.0 _ 4 _ 1	
SyR _ 1.0.0 _ 5		SyR _ 1.0.0 _ 5 _ 1	
SyR _ 2.0.0 _ 1		SyR _ 2.0.0 _ 1 _ 1	
SyR _ 2.0.0 _ 2		SyR _ 2.0.0 _ 2 _ 1	
SyR _ 2.0.0 _ 3		SyR _ 2.0.0 _ 3 _ 1	
SyR _ 2.0.0 _ 4		SyR _ 2.0.0 _ 4 _ 1	
SyR _ 2.0.0 _ 5		SyR _ 2.0.0 _ 5 _ 1	
SyR _ 2.0.0 _ 6		SyR _ 2.0.0 _ 6 _ 1	
SyR _ 2.0.0 _ 7		SyR _ 2.0.0 _ 7 _ 1	
SyR _ 2.0.0 _ 8		SyR _ 2.0.0 _ 8 _ 1	

F–10a Bi-directional Traceability at adjacent levels of system hierarchy (extract)

DREAM	Form **Bi-directional Traceability at adjacent levels of system hierarchy**	ID Form **F - 10a** Version **002**

System: **Date:** **Facilitator:**

Participants:

Validated by:

		Status: active A /cancelled C	1. Functional Requirements					2. Effectiveness / Performance Requirements								3. Interface Requirements									
			SyR_1.0.0_1	SyR_1.0.0_2	SyR_1.0.0_3	SyR_1.0.0_4	SyR_1.0.0_5	SyR_2.0.0_1	SyR_2.0.0_2	SyR_2.0.0_3	SyR_2.0.0_4	SyR_2.0.0_5	SyR_2.0.0_6	SyR_2.0.0_7	SyR_2.0.0_8	SyR_3.1.0_1	SyR_3.1.0_2	SyR_3.1.0_3	SyR_3.1.0_4	SyR_3.1.0_5	SyR_3.1.0_6	SyR_3.1.0_7	SyR_3.1.0_8	SyR_3.1.0_9	SyR_3.1.0_10
1. Functional Requirements	SyR_1.0.0_1																								
	SyR_1.0.0_2																								
	SyR_1.0.0_3																								
	SyR_1.0.0_4																								
	SyR_1.0.0_5																								
2. Effectiveness / Performance Requirements	SyR_2.0.0_1																								
	SyR_2.0.0_2																								
	SyR_2.0.0_3																								
	SyR_2.0.0_4																								
	SyR_2.0.0_5																								
	SyR_2.0.0_6																								
	SyR_2.0.0_7																								
	SyR_2.0.0_8																								
3. Interface Requirements	SyR_3.1.0_1																								
	SyR_3.1.0_2																								
	SyR_3.1.0_3																								
	SyR_3.1.0_4																								
	SyR_3.1.0_5																								
	SyR_3.1.0_6																								
	SyR_3.1.0_7																								
	SyR_3.1.0_8																								
	SyR_3.1.0_9																								
	SyR_3.1.0_10																								

F-10b Stakeholder Negotiation “Win-TRIZ”

DREAM	Form **Stakeholder Negotiation "*Win-TRIZ*"**	ID Form **F - 10b** Version **001**

System: ______ **Date:** ______ **Facilitator:** ______

Participants:

Validated by: ______

Win-TRIZ process

1. Enter Win Conditions	**2. Issues that summarize the conflict**	**3. Restate the initial problem**
6. Refers to Altshuller’s Contradiction Matrix	**5. Re,place with physical characteristics**	**4. Separate the opposite physical contradictions**
7. Derive from among the 40 principles	**8. Candidate options**	**9. Stakeholders evaluation**
10. Agreement Schema		

F-11a System Requirements

DREAM	Form **System Requirement**	ID Form **F - 11a** Version **001**

a) SYSTEM REQUIREMENT (SyR)					
ID	SyR	Value	Rational: Why does this requirement exist?	Author	
				ID	Stakeholder

If available paste here the drawing, table, file, site, etc. related to SyR

Applicability and Reuse							
Region		Conuntry		Market segment		Business unit	

b) RELATED STAKEHOLDERS

ID	Stakeholder	ID	Stakeholder	ID	Stakeholder	ID	Stakeholder	ID	Stakeholder

c) SyR MANAGEMENT		
Stakeholder Role	ID	Stakeholder
Assigned to		
Responsible for verification		
Responsible for validation		

d) QUALITY V&V SyR writing and definition

	Is the requirement...?	Yes	No
	Feasible: Is the requirement achievable using existing technologies and manufacturing?	✓	×
	Appropriate to level: Is the SyR appropriate to the level at it is stated?	✓	×
	Complete: Is the SyR explained enough?	✓	×
	Conforming: Is the SyR conform to a standard formal structure?	✓	×
	Verifiable: Can be verified that the system meets or possesses the SyR?	✓	×
	Necessary: Should the system be able to function in the desired way with this SyR?	✓	×
	Singular: Is not the SyR a combination of two or more requirements?	✓	×
	Correct: Is this a SyR that will result in the desired system performance?	✓	×
	Unambiguous: Is there only one interpretation of the SyR?	✓	×
	Implementation free: Does the SyR stament states what is required instead of saying how?	✓	×
	Consistent: Is free of conflicts with other requirements?	✓	×
Validation	**Understanding:** Is this SyR understandable?	✓	×
	Relevance: Do the expression of the requirement allows to define the importance of the solution?	✓	×

e) V&V Set of SyR status

Target release	dd/mm/yyyy	Estimated cost	€	Urgency	low/medium/high

Verification Method

Verification Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status	✓	×	Approval date	dd/mm/yyyy	Verification Results	

Validation Method								
Document review	✓	×	Prototype	✓	×	Model	✓	×
Describe			Describe			Describe		

Acceptance Criteria		
Validation Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status	✓	×	Approval date	dd/mm/yyyy	Validation Results	

Process Control

Corrective Action #	Corrective action	ID	Stakeholder responsible of corrective action

f) TRACEABILITY

Upward traceability: Parent Stakeholder Need (SkN)

ID	SkN	Value	Rational	Author	
				ID	Stakeholder

Upward traceability: Parent Stakeholder Requirement (SkR)

ID	SkR	Value	Rational	Author	
				ID	Stakeholder

Bi-directional traceability: Related SyR

ID	Co-relationship (++ + - --)	SyR	Value	Rational	Author	
					ID	Stakeholder
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					
	(++ + - --)					

g) SkR RISK MANAGEMENT

Risk Identification

Risk Title	
Risk Statement	
Context Statement	
Closure Criteria	

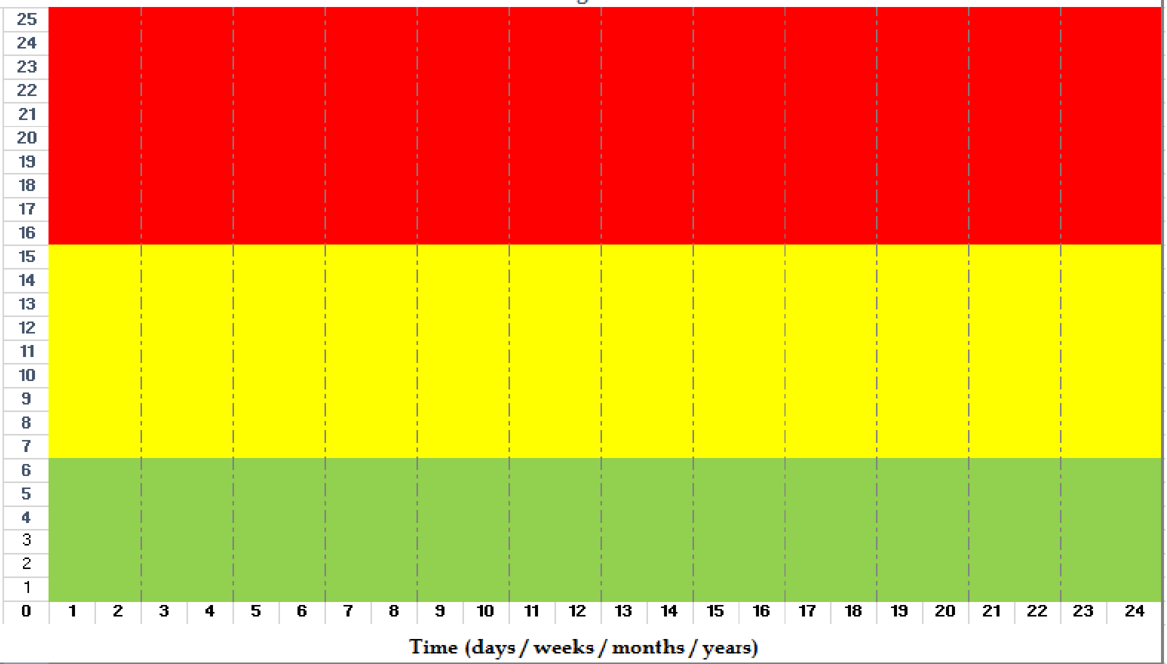
Likelihood	1	2	3	4	5
5	7	16	20	23	25
4	6	13	18	22	14
3	4	10	15	19	21
2	2	8	11	14	17
1	1	3	5	9	12

Consequence

Sunrise	dd/mm/yyyy
Sunset	dd/mm/yyyy
Impact horizont	near/mid/long

Consequence	Rational	Likelihood	Rational

Risk Mitigation



Key Mitigation Action Planned		Success Criteria	IMS No.	Cost	Start	Completion
01					dd/mm/yyyy	dd/mm/yyyy
02					dd/mm/yyyy	dd/mm/yyyy
03					dd/mm/yyyy	dd/mm/yyyy
04					dd/mm/yyyy	dd/mm/yyyy
05					dd/mm/yyyy	dd/mm/yyyy
06					dd/mm/yyyy	dd/mm/yyyy

F-11b Set of System Requirements V&V

DREAM	Form **Set of System Requirements**	ID Form **F - 11b** Version **001**

System: **Date:** **Facilitator:**

Participants:

a) Set of SyR MANAGEMENT		
Stakeholder Role	**ID**	**Stakeholder**
Assigned to		
Responsible for verification		
Responsible for validation		

b) QUALITY V&V Set of SyR writing and definition			
	Is the requirement…?	Yes	No
Verification	**Exhaustive:** Were all the Stakeholders identified and interviewed? Were all the Stakeholder Requirements expressed and writen?	✓	×
Verification	**Complete:** Is the set of SyR explained enough (no need of further amplification)?	✓	×
Verification	**Consistent:** Is the set of SyR free of contradictory individual SkR ?	✓	×
Verification	**Affordable/Feasible: Can the set of** SyR be satisfied by a solution that is feasible within life-cycle constraints?	✓	×
Verification	**Bounded:** Does the set of SyR maintain the identified scope for the intended solution without increasing beyond what is needed?	✓	×
Verification	**Comprehensible:** Are the set of SyR written such that it is clear as: a) to what is expected by the entity, and b) its relation to the system of which it is a part?	✓	×
Verification	**Able to be validated:** Is possible to prove that the set of SyR will lead to the satisfaction of needs while respecting the constraints?	✓	×
Verification	**Non-redundant:** Does NOT the set of SyR have repeated individual SyR?	✓	×
Validation	**Relevance:** the expression of the requirements allows to define the importance of the solution?	✓	×

Note: If any answer is "No", review the set of SyR until all answers are "Yes"

c) V&V Set of SyR status

Target release dd/mm/yyyy

Verification Method

Verification Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Acceptance Status ✓ ×   **Approval date** dd/mm/yyyy   **Verification Results**

Validation Method								
Document review	✓	×	**Prototype**	✓	×	**Model**	✓	×
Describe			Describe			Describe		

Acceptance Criteria		
Validation Metric	Quantity	Units
a) For Approval		
b) For Rejection		

Does the prototype or model meet the stakeholder needs? ✓ ×

Acceptance Status ✓ ×   **Approval date** dd/mm/yyyy   **Validation Results**

Process Control			
Corrective Action #	Corrective action	ID	Stakeholder responsible of corrective action

d) Set of SyR RISK MANAGEMENT

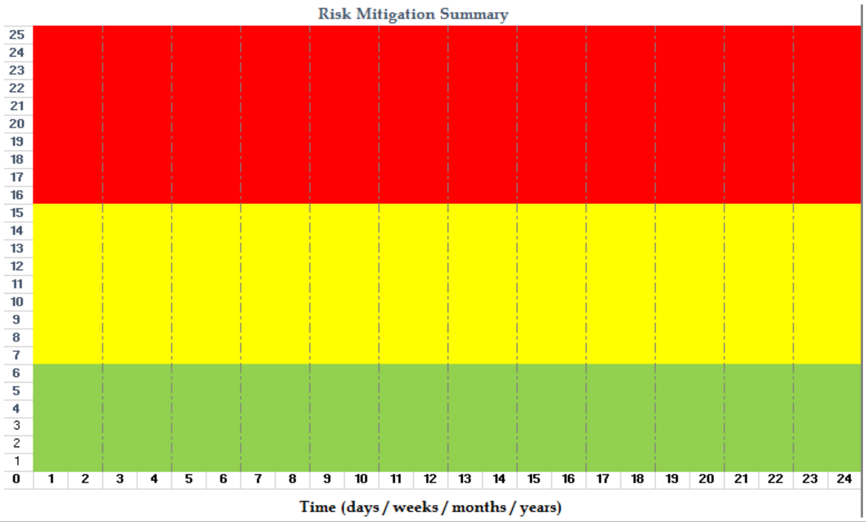
Risk Identification Summary

Risk #	Risk Title

Likelihood					
5	7	16	20	23	25
4	6	13	18	22	14
3	4	10	15	19	21
2	2	8	11	14	17
1	1	3	5	9	12
	1	2	3	4	5

Consequence

External Risk #	Comments



External Risk #	Risk Title	Start	Completion
		dd/mm/yyyy	dd/mm/yyyy
		dd/mm/yyyy	dd/mm/yyyy

Résumé long de la thèse

Méthodologie d'assurance de la qualité pendant la définition des exigences d’un système

Aujourd'hui, il y a une course entre les entreprises pour mettre leurs produits sur le marché (Jakjoud, Zrikem, Baron & Ayadi, 2014; Kiritani & Ohashi, 2015), et cet environnement concurrentiel a forcé l'industrie à optimiser ses processus de fabrication (Jakjoud, Zrikem, Baron & Ayadi, 2013).

Néanmoins, lors de la livraison de nouveaux produits, la connaissance des caractéristiques de qualité qu'un utilisateur final souhaite atteindre est parfois supposée ou mal comprise. Faisandier (2014) précise qu'il n'y a pas nécessairement de liens forts entre les besoins, leur déclinaison en exigences techniques et les solutions potentielles; le résultat est souvent un produit qui n'atteint pas le but pour lequel il a été demandé, conçu et fabriqué (Kiritani & Ohashi, 2015). Actuellement, il n'existe aucune information qui montre explicitement comment les *besoins du client* sont traduits en *exigences du système*, ce qui entraîne une mauvaise qualité du développement du produit (Blanchard & Fabrycky, 2006; Gómez Saavedra, 1991; Gutierrez Pulido, 2010; Kiritani & Ohashi, 2015). Peu à peu, "*les entreprises ont compris qu'il est important de prêter attention à la meilleure façon de concevoir leurs produits*" (Langen, 2002).

Selon Budreau & Boulanger (2014), un *bon* système atteint ses objectifs, justifie sa raison d'être de façon rationnelle, est opérationnel et surmonte les obstacles dans les différentes phases de la vie.

Plusieurs auteurs (Atkinson, 1990; Blanchard & Fabrycky, 2006; Kiritani & Ohashi, 2015) reconnaissent la nécessité d'un effort accru dans la définition initiale des exigences du système, afin d'assurer l'efficacité du processus décisionnel de conception. Kritani & Ohashi (2015), précisent que "*la qualité de la définition des exigences conduit directement au succès ou à l'échec final du projet de développement du système*".

Selon Oehmen (2012), le thème "*exigences instables, imprécises et incomplètes*" occupe la deuxième place parmi les "*thèmes de défis dans la gestion des programmes d'ingénierie*", après "*extinction des incendies - exécution réactive du programme*", car il affecte sérieusement l'efficacité et l'efficience du programme. Voici quelques exemples de problèmes liés aux "*exigences instables, imprécises et incomplètes*": compréhension incomplète ou erronée des exigences des parties prenantes, les exigences dérivées ne sont pas identifiées, les exigences ne sont pas formulées correctement, les exigences sont peut-être en conflit les unes avec les autres et les perceptions des parties prenantes sur la valeur ne sont pas claires. Badreau & Boulanger (2014) ajoutent que les principales causes d'échec du projet liées aux exigences sont les suivantes : les exigences sont incomplètes, les utilisateurs ne participent pas au projet, les attentes sont irréalistes et la gestion des exigences est inefficace lorsque les exigences changent au cours du projet.

Batra & Bhatnagar (2017) ont mené une étude afin d'évaluer huit méthodes de processus d'ingénierie des exigences. Ils ont considéré les caractéristiques de qualité souhaitables présentes dans les modèles comme: la linéarité, le support des exigences changeantes, la nature itérative, le support de l'ingénierie inverse, l'évaluation des risques, les critères pour les exigences spécifiques des applications et les techniques de détermination des exigences, de prétraitement des exigences, de priorisation des exigences et l'estimation des efforts. Leurs conclusions soulignent que "*les chercheurs ont fait des progrès considérables dans le domaine de l'ingénierie des exigences, mais qu'il reste encore du travail de développement à faire... d'autres travaux sont nécessaires comme le prétraitement des exigences, la gestion des risques, la priorisation des exigences, l'application de techniques spécifiques d'incitation, etc*." (Batra & Bhatnagar, 2017). Selon leur étude, le modèle d'ingénierie des exigences en spirale de Kotonya & Sommerville semble être le meilleur parmi d'autres; néanmoins ce modèle ne remplit que cinq des dix caractéristiques souhaitées.

De plus, des auteurs comme Walton (1999) affirment que les *leçons apprises* pour le développement de chaque nouveau produit devraient être saisies, documentées et appliquées au développement de produits futurs, mais qu'il n'existe pas de méthodologie qui puisse servir de guide pour le faire. Des auteurs comme Baines et al. (2006) ajoutent que les activités du processus de conception de produits qui ajoutent vraiment de la valeur n'ont pas encore été résolues, mais il y a l'intention de stabiliser et d'appliquer des techniques spécifiques pour réussir dans la conception de produits allégés.

Connaissant tous ces faits, une question de recherche se pose et motive ce travail de recherche.

Est-il possible d'orienter méthodologiquement les processus d'identification des besoins des parties prenantes et de les traduire en exigences système pour assurer, dans la mesure du possible, la qualité d'un produit ?

L'objectif général de ce travail de recherche est centré sur le développement d'une méthodologie et de ses outils, basés sur l'ingénierie des systèmes et conformes à la norme de qualité ISO 15288 (2015), pour s'assurer, dans la mesure du possible, que les besoins des parties prenantes sont compris, et ensuite traduits en exigences système, réduisant les risques d'avortement ou de retards des projets, et ajoutant de la valeur au processus de conception et d'analyse.

Les résultats escomptés sont ciblés :

1. Une méthodologie et ses outils pour s'assurer, dans la mesure du possible, que l'équipe d'analyse et de conception a compris les besoins des parties prenantes et les a traduits en exigences du système.
2. La validation de la proposition par le biais d'une étude de cas (ou plus), générant une documentation qui démontre le suivi de la méthodologie, étape par étape, de la traduction des besoins des parties prenantes en exigences système.

L'objectif peut être atteint par l'intégration de différents domaines, techniques et méthodologies. L'application d'outils et de méthodologies de qualité réduira le temps pendant lequel un nouveau produit sera disponible pour l'utilisateur, ce qui assurera son efficacité et son efficience et rendra l'organisation plus compétitive (Bauch, 2004). La philosophie Lean Thinking fera fonctionner les processus comme une horloge (Oehmen, 2012). L'assurance de la qualité surveillera les processus jusqu'à la prévention des erreurs et au respect des exigences

(ISO 9000, 2005), tandis que l'ingénierie des systèmes s'occupera du cycle de vie des produits, à partir de l'identification des besoins jusqu'aux étapes de retrait des produits (Blanchard & Fabrycky, 2006; Ryan & Faulconbridge, 2016; Faisandier, 2014).

Ce travail de recherche a l'originalité d'intégrer différents domaines et plusieurs contributions d'auteurs pour accompagner les organisations dans la phase de conception du cycle de vie du système, principalement sur l'identification des besoins des parties prenantes et la définition des exigences du système, y compris la génération de ses documents - à des fins de gestion qualité - conformément à la norme ISO 15288 (2015) et, en même temps, d'obtenir un produit ou service qui répond à ces besoins.

Notre recherche est basée sur la discipline de l'ingénierie des systèmes, par conséquent, le contexte du problème, le centre d'intérêt de la recherche, les normes de qualité à des fins de gestion de la qualité et l'assurance qualité des produits sont étroitement liés à l'ingénierie des systèmes.

Dans ce contexte, l'analyse documentaire couvre l'ingénierie des exigences, considérée comme une discipline de l'ingénierie des systèmes, afin de définir ce qu'est une *exigence*, comment exprimer un *bon* énoncé d'une exigence, ses attributs et ses classifications, les différents modèles de processus d'ingénierie des exigences et l'importance de la traçabilité des exigences et sa gestion pendant tout le cycle de vie du système.

Comme le premier résultat attendu de ce travail de recherche est de *comprendre les besoins des parties prenantes et de les traduire correctement dans les exigences du système*, ce travail de recherche aborde spécifiquement cette question. La traduction des besoins des parties prenantes en exigences du système peut se faire en deux étapes successives, d'abord *traduire les besoins des parties prenantes en exigences des parties prenantes*, puis *traduire les exigences des parties prenantes en exigences du système*. Il est important de souligner que ce travail de recherche est basé sur la révision bibliographique de systèmes complexes en général.

La documentation est abondante sur la façon de mener ces deux processus, leurs activités et les tâches connexes. Nous avons profité de l'occasion pour analyser et intégrer toutes les contributions pertinentes dans un processus bien défini pour aider à résoudre le problème. Notre première conclusion de cette analyse est la suivante: malgré les nombreuses contributions trouvées dans la littérature, nous n'avons pu trouver aucun indice sur la façon d'effectuer certaines activités ou tâches. Par exemple, il n'existe aucune méthode objective pour évaluer le niveau d'importance des parties prenantes, ni aucune méthode structurée qui aide à déterminer ses besoins, ni même une définition claire de ce qu'est la *traçabilité bidirectionnelle aux niveaux adjacents de la hiérarchie du système*. Ce sera donc un point que nous aborderons dans notre recherche.

En outre, il est encore possible de concevoir un processus *lean* pour traduire les besoins des parties prenantes en exigences système qui ajoutent de la valeur tout en réduisant les déchets. Pour cette raison, nous avons décidé de nous pencher sur le Lean Thinking afin d'aborder le sous-objectif de *réduire les risques d'avortement ou de retards du projet, et d'ajouter de la valeur au processus d'analyse et de conception*.

Nous avons passé en revue la philosophie Lean Thinking, ses concepts fondamentaux, la façon dont cette philosophie a été appliquée pour permettre le *Lean* dans les phases du cycle de vie du système (Ingénierie des Systèmes), et la façon dont le *Lean* est mis en œuvre dans les systèmes de gestion de la qualité (assurance qualité).

A ce stade, compte tenu des objectifs de la recherche, des leçons tirées de l'analyse de l'état de l'art et des conclusions tirées de ces analyses, nous en arrivons à certaines propositions de contributions.

Nous avons développé une méthodologie résultant de ce travail de recherche qui répond à l'objectif *d'assurer, dans la mesure du possible, que les besoins des parties prenantes soient compris et ensuite correctement traduits en exigences du système, réduisant les risques d'avortement ou de retards du projet, et ajoutant de la valeur au processus d'analyse et de conception*. Nous avons appelé cette méthodologie *DREAM*, pour *Driven Requirements Analysis Management* (gestion guidée de l'analyse des exigences).

DREAM a été conçu à partir des contributions de plusieurs auteurs qui nous permettent de construire une proposition plus forte que les existants, complète, couvrant toutes les tâches nécessaires à la réalisation des deux processus (des besoins des parties prenantes aux exigences des parties prenantes, et des exigences des parties prenantes aux exigences système). Pour certaines activités, nous avons analysé l'information disponible pour décider entre différentes options quant à la façon de mener les activités ou les tâches; dans d'autres cas, il n'y avait pas de chemin ou d'indice clair à suivre; par exemple, il n'y a aucune méthode objective pour évaluer le niveau d'importance des parties prenantes, ni aucune méthode structurée permettant de déterminer les besoins, ni même de définir clairement la *traçabilité bidirectionnelle aux niveaux adjacents de la hiérarchie du système.* Néanmoins, une faiblesse est toujours une opportunité; le fait qu'il n'y avait pas de piste pour mener à bien ces activités nous a permis de donner libre cours à notre créativité pour faire des propositions intéressantes.

DREAM se concentre sur le soutien des deux étapes de la traduction des besoins des parties prenantes en exigences du système : *des besoins des parties prenantes aux exigences des parties prenantes* et *des exigences des parties prenantes aux exigences du système*. De plus, nous avons conçu DREAM comme un système de gestion de la qualité (SGQ) qui, après avoir formé les parties prenantes, peut être mis en œuvre dans une organisation. DREAM en tant que SGQ contient différents niveaux de documents, comme des *processus*, des *instructions de travail* et des *formulaires* conformes à la norme ISO 15288 (2015). Une fois ces formulaires remplis, ils deviennent des documents vérifiables ou *records*, ce qui permet de réutiliser ces informations dans d'autres projets.

Une fois DREAM élaboré, il était nécessaire de vérifier s'il répondait à la question de recherche : *Est-il possible d'orienter méthodologiquement les processus d'identification des besoins des parties prenantes et de les traduire en exigences système pour assurer, dans la mesure du possible, la qualité d'un produit ?*

En outre, nous devions également vérifier si DREAM avait atteint l'objectif de fournir une *méthodologie et des outils, basés sur l'ingénierie des systèmes, conformes à la norme de qualité ISO 15288 (2015), pour nous assurer que les besoins des parties prenantes sont compris et ensuite traduits en exigences système*.

La vérification a été effectuée que *DREAM comme un SGQ* est conforme à la norme ISO 15288 (2015) et, en raison de la nature de la recherche, la validation a été effectuée selon deux méthodes qualitatives :

a. *Etude de cas*: l'application de DREAM comme un SGQ nous a permis :
   i. conceptualiser des systèmes qui valident que les besoins des parties prenantes ont été compris et, plus tard, traduits en exigences du système ; et
   ii. produire de la documentation (records) qui démontre le suivi de la méthodologie, étape par étape
b. *Questionnaires* pour obtenir une rétroaction en vue d'évaluer la proposition et de l'améliorer par la suite.

Les résultats confirment que *DREAM* est totalement conforme à la norme ISO 15288 (2015).

De plus, *DREAM* est validé par les professeurs et les étudiants qui l'ont mis en œuvre dans des projets étudiants ; leur engagement et leur aide sont inestimables car les étudiants, qui n'avaient aucune expertise dans la conceptualisation de nouveaux systèmes, ont trouvé DREAM utile comme *guide méthodologique pour identifier les besoins des parties prenantes et les traduire en exigences système pour assurer, autant que possible, la qualité du produit*.

Comme indiqué précédemment, Batra & Bhatnagar (2017), exprime qu'il y a certaines caractéristiques de qualité souhaitées qu'une méthode de processus d'ingénierie des exigences devrait avoir; la méthodologie DREAM satisfait neuf de ces dix caractéristiques de qualité souhaitées: 1) elle est linéaire, 2) elle soutient l'évolution des besoins, 3) elle a un caractère itératif qui permet 4) la rétroaction des utilisateurs, l'évolution des besoins et les besoins dérivés; 5) la rétro-ingénierie est effectuée par l'application du questionnaire pour obtenir les besoins, 6) ce questionnaire basé sur la classification des exigences d'un système est une technique spécifique pour obtenir des besoins; 7) elle permet également l'analyse des risques, 8) les besoins des parties prenantes sont analysés avant qu'ils ne deviennent des exigences, c'est-à-dire, puisqu'il s'agit des besoins des parties prenantes; et 9) les exigences sont priorisées automatiquement en fonction de l'ordre de priorité des besoins. La caractéristique de qualité manquante que DREAM ne satisfait pas (selon Batra & Bhatnagar, 2017), est l'"estimation de l'effort" lors de son application dans un projet.

À cet égard, les élèves ont trouvé que *DREAM* est assez difficile à appliquer à certaines étapes. Ils ont suggéré que l'automatisation de *DREAM* permettrait de réduire la complexité des formulaires proposés, de gagner du temps et d'économiser des efforts ; ils aimeraient également avoir une formation préalable avant d'appliquer *DREAM*.

En conclusion, on répond à la question de recherche et les résultats escomptés ont été atteints :

1. *DREAM* et ses outils s'assurent, dans la mesure du possible, que l'équipe d'analyse et de conception a compris les besoins des parties prenantes et les a traduits en exigences système.
2. La proposition a été validée par le biais de plusieurs études de cas d'étudiants, développés à Tecnológico Nacional de México, Instituto Tecnológico de Toluca (ITTol), Mexique, générant une documentation qui démontre le suivi de la méthodologie, étape par étape, de la traduction des besoins des parties prenantes en exigences du système.