Gestión y tecnología para la ingeniería de requerimientos en servicios computacionales

Management and technology for engineering requirements in computer services

Engenharia de gestão e requisitos de tecnologia em serviços de informática

Luis Ambrosio Velázquez García
Universidad Autónoma de Querétaro, México
luismxbon@hotmail.com

Resumen

El presente artículo describe el enfoque teórico sobre la gestión en la ingeniería de requerimientos que toda organización necesita a través de la incorporación de la tecnología, para la toma de decisiones sobre qué método o modelo se empleará, ya que desempeña un papel primordial en el proceso de producción de software dado que se enfoca en un área fundamental: la definición de lo que se desea producir.

La principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas. Debido a ello en este trabajo se exponen dichos métodos o modelos con sus características, técnicas y/o herramientas a tomar en cuenta para el buen desarrollo de software y su implementación.

Esta investigación es documental y se concluyó que un factor relevante en la gestión y tecnología para la ingeniería de requerimientos es la viabilidad de implementar un software mediante el análisis de requerimientos y su especificación formal, lo cual reducirá costos y problemas en la operación.

Palabras clave: gestión industrial, elección de tecnología, aplicación informática.

Abstract

This article describes the theoretical approach on the management in Requirements Engineering that every organization needs through the incorporation of technology, for decision making about which method or model will be used, Since it plays an essential role in the process of software production given that focuses on a fundamental area: the definition of what you want to produce.

The main task consists in the generation of correct specifications describing clearly, unambiguously, in way consistent and compact, the needs of users or customers; in this way is intended to minimize the problems of poor requirements management in systems development. As a result in this work are exposed such methods or models with their characteristics, techniques and tools to consider for good software development and implementation.

This documentary research concluded that a relevant factor in management and technology for requirements engineering It is the feasibility of implementing a software requirements analysis and its formal specification, which will reduce costs and problems in operation.

Key Words: industrial management, choice of technology, computer application.

Resumo

Este artigo descreve a abordagem teórica da engenharia de requisitos de gestão que toda organização precisa através da introdução de tecnologia, para tomar decisões sobre qual método ou modelo será usado como ela desempenha um papel fundamental no processo de produção de software, uma vez que se concentra em uma área chave: a definição do que deve ser produzido.

A tarefa principal consiste em gerar especificações corretas que descrevem claramente, de forma inequívoca, coerente e compacto, as necessidades dos usuários ou clientes; Esta abordagem destina-se a minimizar os problemas de má gestão dos requisitos no desenvolvimento do sistema. Como resultado deste trabalho, esses métodos ou modelos com as suas características, técnicas e / ou ferramentas para ter em conta para o bom desenvolvimento de software e sua implementação são discutidas.

Esta pesquisa é documental e concluiu que um fator relevante na engenharia de requisitos de gestão e tecnologia é a viabilidade de implementação de um software de análise de requisitos e especificação formal, o que reduzirá os custos e problemas operacionais.

Palavras-chave: gestão industrial, escolha da tecnologia, aplicação de computador.

Fecha recepción: Diciembre 2015 **Fecha aceptación:** Julio 2016

Introduction

The requirements are the basis for all projects involving software development. The definition of users, customers, suppliers, developers and companies involved in the creation of a new system, will result in what the system must do to meet the needs of the project.

A large number of software projects fail for not doing a proper definition, specification, and requirements management. Within that maladministration factors can be found as the lack of user participation, incomplete requirements and the poor requirements change management.

By agreeing to the requirements for the system, will provide the basis for development planning of a software project. These must be understood by all those involved in the project, expresados en un lenguaje común, capturing the problem needs fully, unambiguous.

The requirements allow risks management, even since the beginning of the system development. The risks posed can be tracked, its impact evaluated and effects of the plan mitigated by generating big savings on the project.

Requirements engineering serves to determine the needs of a client in the implementation or development of software, through the process of collecting, analyzing and check in a way complete and correct software requirements specification (Chaves, 2011). In addition, it

contributes to build a software product of high quality, under the constraints of time and budget, requiring rigor, creativity, documentation and management in all its activities (Pohl, 2013).

The requirements are essential in projects involving software development, because they cover such activities as planning, which refers to the estimates of time and cost; these also considered the definition of the necessary resources and the preparation of schedules, that will be one of the main mechanisms of control that we can rely on during the development stage.

Generally it costs representing software development are often greater than the hardware. Engineering Requirements (IR) has to do with that development is done in a way that is economically viable. When customers, users and software developers do not know precisely the problem they want to solve, chances are that a useful and optimal solution is obtained.

You need to know and get the requirements correctly because they are a critical point in the development of software, regardless of the type of development concerned. A misperception or documentation thereof lead to a lot of problems that arise in the development life cycle.

There are techniques, tools and models that have a primary role in the software production process and focusing on a key area: the definition of what is to be produced. Its main task is generating correct specifications describing clearly, unambiguously, consistent and compact, the needs of users or customers; that way is to minimize problems by poor management of requirements in system development.

Because of this this work aims to show the techniques, tools and models to consider and propose a good software development.

THEORETICAL FRAMEWORK

Basics of Requirements Engineering

The theoretical perspective of this research takes up the principles of studies by Leite (1987), Sommervile (2005), Arias (2005), Jeffries (2001), Insfran (2014), among others, which describes some of the most recognized definitions on Requirements Engineering (IR).

For example, to Leite (1987) requirements engineering is the process by which different views are exchanged to collect and model what the system will perform. This process uses a combination of methods, tools and actors, whose product is a model from which a requirements document is generated.

Or a condition or need for a user to solve a problem or achieve a goal, or a condition or capability that must be present in a system or system components to satisfy a contract, standard, specification or other formal document (IEEE 1990) .

For Sommerville et al. (2005) is a requirement or just an abstract high-level declaration of a service to be provided by the system or a restriction.

Arias (2005) states that the requirements are essential in projects involving software development. Cover activities such as planning, referring to estimates of time and costs, also consider the definition of the necessary resources and developing schedules to be one of the main mechanisms of control that is considered during the development stage.

Jeffries et al. (2001), say that a story is not only a description of functionality normally exposed in a post-it, but a user story is also formed by two parts:

1. The conversation involved, since it is a tool to interact; and
2. The way your implementation, testing and verification are confirmed.

The functional requirements are those that define the functions that the system will be able to perform as they describe the transformations that the system performs on the inputs to produce outputs. It is important to describe the what and not how those changes should be made. These requirements, while the software project progresses, become algorithms, logic and much of the system (Insfran, 2014) code.

User stories are used to define the requirements of a software system and also to create estimates for planning interactions. These are written by customers in the form of things they want the system to do for them (Jeffries et al., 2001).

These cards are a technique used to specify the software requirements. It is paper cards in which the client briefly describes the features that the system must possess, whether functional or non-functional requirements. In turn software requirements can be divided into 2 categories: 1) functional requirements and 2) non-functional requirements (ibid, 2001)..

Requirements show whether or not achieve the objectives set out in the project and they are a standard of the needs of customers or users of the system, as will be checking to see if they are meeting the goals (Arias, 2005).

Insfran (2014) He mentions that it is important to know that a requirement should be:

- Specified in writing, it ie any contract or agreement between two parties.
- Possible to test or check, because if a request can not be verified, then how do you know if it is fulfilled or not?
- Concise, a requirement is concise if it is easy to read and understand. Its wording should be simple and clear to those who will consult it in the future.
- Complete, if the request is complete, if need further details in its wording, ie, if sufficient information is provided for your understanding.
- Consistent, ie, a requirement is consistent if it does not contradict other requirements.

- Not ambiguous, for example, a requirement is unambiguous when it has only one interpretation. The language used in its definition should not cause confusion to the reader.

During the stage of requirements specification can present many disadvantages, which are important to identify and prevent. Here is a list that is presented Sommervile et al. (2005) provides for the most common problems in this process:

- The requirements are not obvious and come from many sources.
- They are difficult to express in words (the language is ambiguous).
- The number of requirements in a project can be difficult to handle.
- A requirement may change throughout the development cycle.
- You can not explain what it does.
- Tends to remember the exceptional and forget the routine.
- Talk about what does not work.
- Users have a different vocabulary developers.

Models, methods and / or tools that describe Requirements Engineering

The different models, methods and / or tools that describe requirements engineering in organizations are exposed. There are several alternatives to define the optimal characteristics that must have a good user story card. Some of these are:

The 3 "C"

A user story describes the desired functionality from the client's perspective, describing the functionality, who needs it, how and why to use. The basic characteristics of a user story can be summarized in three elements known as the 3 "C" of a user story by its initials in English *Card, Conversation, Confirmation* (Jeffries, 2001):

1. Card (Card) is the written history, which serves as identification, reminder and also helps plan description.
2. Conversation (Conversation) is the core of the story, dialogue happens with users, notes, recordings, prototypes and documents.

3. Confirmation (Confirmation) is the criterion for acceptance tests that the user will use to confirm that the story was finished.

Another important feature is represented by the acronym SMART INVEST and (by its acronym in English) and long are marked as a reference for the realization of a good story card.

The model INVEST

Wake (2003) INVEST suggested model, that is, an Independent, Negotiable, Estimable, Small and Verifiable history. These elements are described below:

- Independent (Independent): a story should be independent of others. Dependencies between stories makes it more difficult to plan, prioritize and estimate. Although you can not always do this, you can reduce dependencies by a combination of stories, or based stories differently.
- Negotiable (Negotiable): a user story is negotiable. It is not an explicit contract must contain features that system development. Card history is only a short description that does not include details.
- Valuable (Valuable): every story has to have value for the customer. A very good way to generate valuable stories is to make the customer type.
- Estimable (estimable): developers need to estimate a user story to allow it to prioritize and plan history. an accurate estimate is needed, but just enough to help customer requests and plan the implementation of what is on the card history. a function of size is also involved; greatest stories are harder to estimate.
- Small (Small): a good story should be small in effort, generally representing no more than 2-3 people / work week. smaller stories tend to get more accurate estimates.
- Verifiable (Testable): a history needs to be testable stage confirmation occurs. If a customer does not know how to try something, this may indicate that the story is not clear enough, or does not correspond with something valuable to them, or that the customer only needs help in testing.

The attributes that carries the INVEST method are: the feedback loop of the proposal, estimation and implementation, which help significantly to achieving cards user story way and teach the team what each should to do for the software development project does not fail.

Model SMART

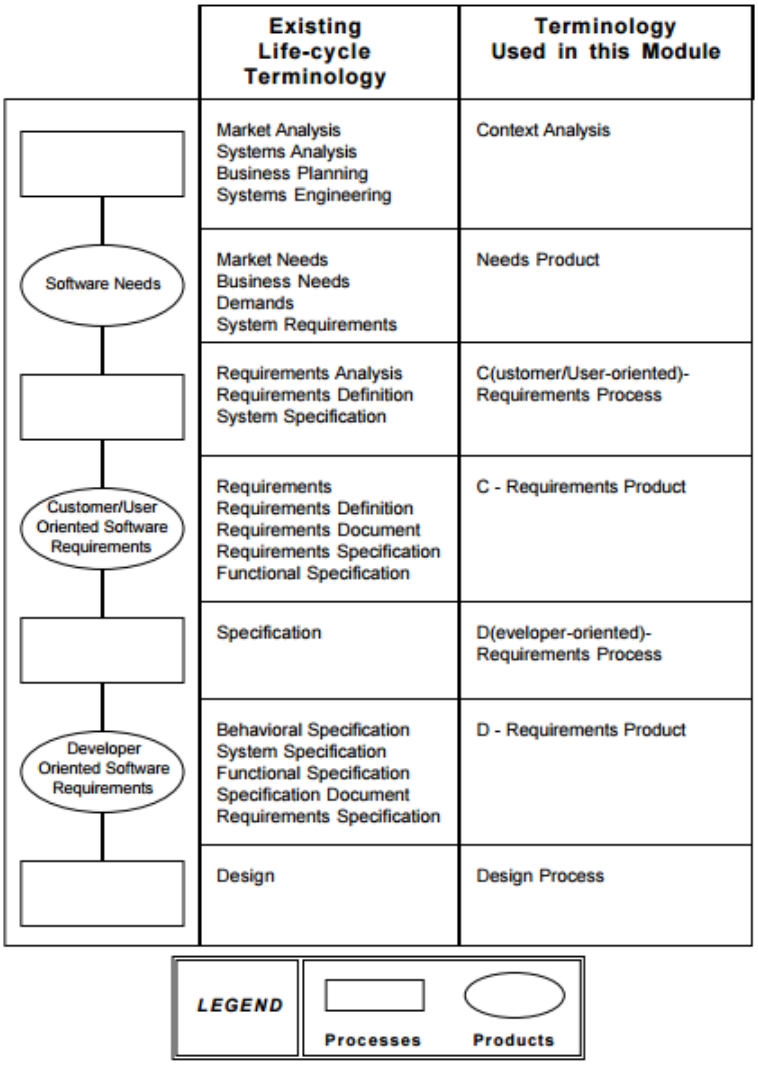
As Wake (2003) mentions, the task-based Smart or SMART model includes specific, measurable, achievable, relevant and time box must have a user story card. It is defined by the following:

- Specific (Specific): a task has to be specific enough so that the entire group can understand what is reflected in it.
- Measurable (Measurable): can say is already finished? It should be measurable and understand that the team has to agree that the process does what it is intended, the tests are completed and the code has been programmed.
- Achievable (Archiveable): the programmer should be able to perform the task that is asked. There is a rule in the development team where anyone can ask for help when you need it, which certainly ensures that the task is completed.
- Relevant (Relevant): each task must be relevant and contribute significantly to the development of the project way. The stories are divided into tasks for the benefit of developers and end them explain and justify to the client.
- Box time (Time - Boxed): a task is limited to a specific duration. It can be estimated in hours or days, but if you can not perform there must be an expectation of when to ask for help. If a task is more complicated than expected, the team should know divide the task, change the developer or do something to help perform the task in the estimated time.

For Brackett (1990), the elements that are integrated into the terminology lifecycle shown in Figure 1 are given when they have determined the general system requirements and has taken the decision that certain functions are performed by software:

1) The process should begin with defining requirements and analyze the context of that question, ie a brief description of the need,
2) Define Necessity: this includes external elements of the company such as the market, the company's internal needs, real and potential demand, as well as the specific requirements of systems that help decisions in the company,
3) Define the requirements process, ie, avoid too much detail, flexibility must be present so that the team can adjust how much of the user's need is to be implemented,
4) Identify product requirements is to be able to further develop in any sequence and that the team can use to plan,
5) Define the process requirement oriented developer,
6) Delineate product requirements (specification of end-user behavior and system) and finally
7) The design process will involve analyze and test acceptance criteria document or list for history in defining the life cycle.

Figure 1. Terminology used to define the lifecycle



Source: Brackett (1990).

Basics of management and technology

On the other hand, it is necessary to define the role of management and technology, to apply requirements engineering, as such concepts from the contributions of Valencia (1996), Mora (1999), analyzed BID- SECAB-CINDA (1990), and Zoltan (1995).

Overall, management concepts, management and management are synonymous, despite the great efforts and discussions that have been made to differentiate between them. In practice it is noted that the term has been translated as management administration, but also as management.

In this conception, it is for the manager to look at the environment so that the organization can generate development: take resources and produce more resources. Administrator accounts for more maintenance and upkeep, while the administration is conceived functional or vertical (Valencia, 1996).

Mora (1999) stated that management is linear or traditional where it is synonymous with management. For management is the set of steps that are performed to develop a process or to achieve a particular product. It is assumed as the direction and government activities to make things work, with capacity to generate processes of transformation of reality.

On the other hand, the IDB-SECAB-Cinda (1990) gives the following ranking of technology according to the degree of incorporation and the degree of modernity:

- Degree of Incorporation:
- Hardware: the built-in machine technology.
- Software: is unincorporated and is presented through magazines, books, manuals, videos, computer software technology.
- Orgware: organizational structures.
- Humanware: is the built-in people who have a "know-how".
- b) Degree of modernity:
- Primitive Technology: corresponds to the primitive, slave and feudal times.
- Modern technology: In contrast to the primitive is produced in recent decades.
- Backward technology: is that "it has been surpassed in some factor" for example, the electric typewriter with regard to the memory writing.
- Cutting edge technology: is one that has just been produced. It is "fresh from the oven".

Being multi and interdisciplinary, technology management itself can not be subject to any profession or discipline (Zoltan, 1995). It is a process that deals with the interfaces between science, engineering, economics and management institutions.

Technology management promotes the organization and execution of tasks closely with the agents (researchers, engineers, scientists, technologists), but is not itself, nor does it contain within it the conduct of research, innovation or control quality (ibid., 1995).

In the glossary of terms IDB-SECAB-Cinda (1990), indicates that technology management is the discipline in which knowledge of engineering, science and management are mixed in order to perform planning, development and implementation of technological solutions that contribute to achieving the strategic and technical objectives of an organization.

According to Zoltan (1995), technological management is at the level of a country. Since this is in the global technology market, foreign customers of domestic enterprises but also competition are government policies, as well as a set of international rules on trade, investment, quality, environment, patents, among others.

On the other hand, within the country they are: a government with policies on science, technology and education; a national technology market and finally companies with customers and competitors and all the national and global context must take a series of decisions about technology. They must decide whether to make its own development or seek technology abroad; in the first case they should make innovation in the second evaluate, select, negotiate, buy, adapt, among all those activities that require a whole concept of strategic planning (ibid., 1995).

METHODOLOGY

Then the general objective and specific objectives, as well as the hypothesis and the method is presented to provide an overview to describe the process of the investigation.

General objective

Ask a process of Requirements Engineering through Management and Technology for comparing requirements, techniques, tools and models for application in a business computer services, to satisfy the needs of the conditions commonly omitted development requirements.

Specific objectives

- Get information to the requirements, techniques, tools and models Requirements Engineering.
- Perform comparative characteristics.
- To demonstrate what are the advantages, disadvantages, importance and scope.

Hypothesis

It is intended to outline a procedure requirements engineering, characterized by its importance and scope to the user, considering the best results of the approaches discussed and supported by various methods, techniques and tools already developed. It also seeks to unify the terminology used in the field of Requirements Engineering, eliminating the differences that may arise as a result of semantic confusion and improve the process of it.

Therefore, the following hypothesis is proposed:

H_i: If a company applies computer services Engineering requirements, then efficiency will be achieved in the process.

H_o: If a company does not apply computer services Engineering Requirements, then there will be efficiency in the process.

Method

The research was essentially constitutes the qualitative-documentary approach, since the population and sample consists of documents and constitutes a strategy where it is observed and systematically reflect on realities using for this different document types (Gallant, 2011). The documentary population is made up of references of use and universal scientific knowledge, basically specialized national and international scientific articles. Examines, interprets, presents data and information on a given subject of any science, using analysis and whose purpose is to obtain results that can serve as a basis for the development of scientific creation (Cortés, 2004).

Discussion and conclusions

It is necessary to prioritize, define and implement processes requirements engineering a formal process with clearly defined activities and products, where a document containing the results of this stage for the proper development of the project to be implemented is also located.

When there is no document containing the formal processes, they can not be properly contemplate the costs, development times, risks and estimates that arise during the life of the project, and therefore the requirements do not materialize and the product you want to get fails or does not meet customer expectations and primary developer of the same.

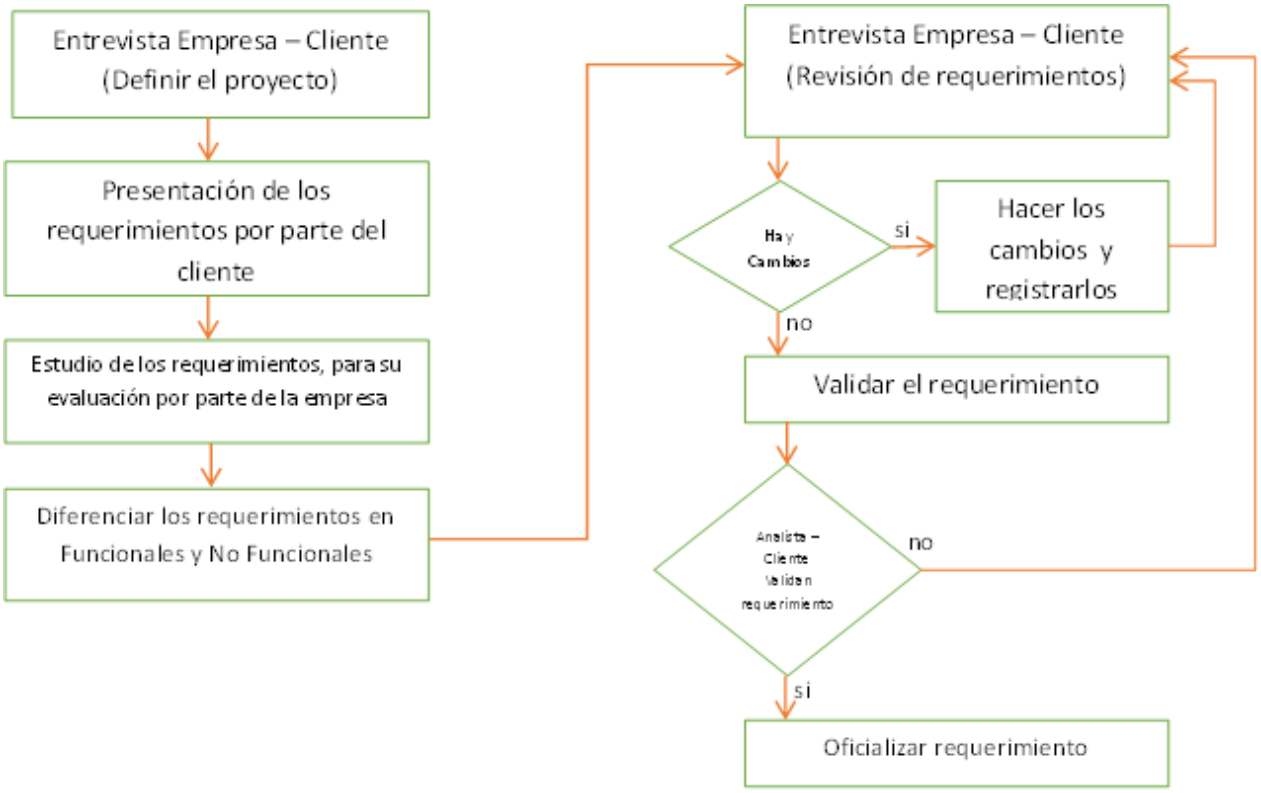
It requires documentation and maintenance requirements for a specific software product taking into account the contribution of Requirements Engineering.

And determining the feasibility of carrying out the software by obtaining and analyzing requirements and formal specification.

It is necessary to define the procedure through Requirements Engineering Management and Technology as a basic element for the fulfillment of the objectives set in any project to implement in the organization. The phase of the validation of the proposed procedure should be performed by applying a real case study applied to the organization by customer approval.

a proposal that starts with the process of Requirements Engineering to compare the requirements, techniques, tools and application models in a computer services company as part of the analysis in this research is presented in figure 2 below.

Figure 2. Procedure IR



Source: Creación propia (2016), a partir de Arias (2005), Jeffries (2001), Insfran (2014), Baez (2001), Beck (2000), Cohn (2009), Mazan (2010) y Rzepka (1989).

This proposal generated as a result of theoretical analysis, contains the design of common activities and the format to be held for the project, which will incorporate features of the methodologies of engineering requirements, always seeking to meet the needs of conditions commonly omitted in the development of requirements.

It is concluded that in any organization want to implement through technology management and engineering requirements, due in the first interview (between the company offering the service and client) define the project, expectations and scope; as well as the impact within the organization and externally.

Once the project is exposed must be submitted customer requirements, these are analyzed by the offeror company and determines which are functional and nonfunctional.

We should make a second interview where the requirements are revised, the changes are diagnosed (if any) of each of them and validated. Once validated by both parties, these are formalized and start working on that request.

The use and performance of a good model applied in Requirements Engineering is used to identify and define all the activities involved in the discovery, documentation and maintenance requirements for the successful development of a project and help determine process feasibility to see if this is feasible or not.

By getting the tools that allow the project is well documented, and where both the client and the developer use the same language, a specification formally achieved, obtaining a thread validation verifies that really asking the customer defines the system being developed (Sikora et al. 2012).

Bibliography

Arias, M. (2005). La Ingeniería de Requerimientos y su importancia en el desarrollo de proyectos de software. *Revista Intercedes*, Edición Digital, Volumen VI, Número 10.

Baez M., Barba S. (2001). Metodología DoRCU para la Ingeniería de Requerimientos. Instituto Superior Politécnico "José Antonio Echeverría", La Habana, Cuba.

Beck K., Folwler, M. (2000). Planning extreme programming. USA: Addison Wesley Pub Co Inc.

BID-SECAB-CINDA (1990). Glosario de términos de gestión tecnológica. Colección Ciencia y Tecnología Nº. 28. Santiago de Chile.

Brackett, J. (1990). Software Requirements, SEI Curriculum Module SEI-CM-19-1.2.

Chaves, M. A. (2011). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. InterSedes, 6(10).

Cohn, M. (2009). User stories applied for agile software development. 13th printing. Indiana, USA: Pearson Education, Inc.

Cortés C. M., Iglesias L.M. (2004). Generalidades sobre Metodología de la Investigación. ISBN: 968-6624-87-2. Editorial Universidad Autónoma del Carmen, Colección Material Didáctico: México.

Galán, A.M. (2011). Metodología de la Investigación. Campus virtual UDES.

INSFRAN et al. (2014). Requirements engineering in software product line engineering. Requirements Engineering, 19(4), 331-332.

Jeffries et al. (2001). Essential XP: Card, Conversation, Confirmation. Consultado el 10/06/2013. Disponible en: http://xprogramming.com /articles/expcardconversationconfirm ation/.

Institute of Electrical and Electronics Engineers (1990). IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610. 12-1990 (revision and redesignation of IEEE Std. 729-1983), New York.

Leite, J. (1997). Ingeniería de Requisitos. Notas de Cátedra.

Mazan, M. (2010). ¿Qué son "user stories" y por qué se tiene que "invertir" en ellas? Consultado el 13/06/2013 Disponible en: http://agiland.pe/index.php? view=entry&id=10%3Aique-son-user-stories-historias-de-usuario-y-por-que-se-tiene-que-invertir-en-ellas&option=com.lyften bloggi e &Itemid=29

Mora, Julia (1999). "Transformación y gestión curricular". En: Memorias Seminario Taller Evaluación y Gestión Curricular, Universidad de Antioquía.

Pohl, K. (2013). The three dimensions of requirements engineering. In Seminal Contributions to Information Systems Engineering (pp. 63-80). Springer Berlin Heidelberg.

Rzepka, W. E. (1989). A requirements engineering tested: concept, status, and first results, proceedings of the Twenty-second Annual Hawaii International Conference of Systems Sciences. IEEE Computer Society. Consultado el 20/05/2013. Disponible en: http://ieeexplore.ieee.org/xpl/most RecentIssue.jsp?punumber=233.

Sikora et al. (2012). Industry needs and research directions in requirements engineering for embedded systems. Requirements Engineering, 17(1), 57-78.

Sommerville et al. (2005). Ingeniería del Software. Séptima edición. Madrid, España: Pearson Educación, S.A.

Valencia, Carlos (1996). Gerencia de Proyectos. Seminario para profesores U. de A.

Wake, B. (2003). INVEST in Good Stories and SMART Task. XP123. Consultado el 15/07/13. Disponible en: http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

Zoltan, Szabó. (1995). Seminario sobre Gestión Tecnológica. SENA.