

Requirements Engineering: A Process Model and Case Study to Promote Standardization and Quality Increase

Jose Andre Dorigan
Computer Science Department
State University of Londrina
Londrina, PR, Brazil
jadorigan@gmail.com

Rodolfo Miranda de Barros
Computer Science Department
State University of Londrina
Londrina, PR, Brazil
rodolfo@uel.br

Abstract—In Requirements Engineering, it is very common for requirements to be poorly specified, inconsistent with the client needs or badly written. Based on these problems, this paper presents a model of Requirements Engineering Process for description standardization, through the reuse of words, seeking to improve the specification quality. We present a Case Study to evaluate and identify benefits of its use in an academic software development. Also, a comparative study between processes that deal with requirements quality assurance was developed showing the works difference.

Keywords—Requirements Reuse; Quality on Requirements Description; Requirements Standardization.

I. INTRODUCTION

Requirements are linked to the main problems of software development; correct requirements gathering is one of the most important tasks in software development. In most cases, they do not reflect the real needs of users, because they are incomplete or inconsistent [14][17].

A major difficulty is to ensure that the requirements specification is in accordance with the client ideas [15][18]. Often, there are misinterpretations by the Requirements Engineer (we will use the abbreviation *REng* in this work) or the clients cannot clearly express their real needs [15]. These specification problems create non-standardized and inconsistent requirements.

IEEE standards 830 [4] and 1233a [5] indicate several properties for software and system requirements specification to obtain a good quality level. Some examples of these properties are: how to avoid ambiguity, use of natural language to describe requirements and the possibility of requirements compliance verification.

The process model proposed in this paper is justified on the ideas of requirements description standardization presented in [8]. Utilizing these, we expected a decrease in development time, during the Requirements Specification, an increase in quality description and confirmation of the client needs by validating the requirements created.

Within this scenario, the objective is to propose a model of Requirements Engineering Process to help with description standardization and reuse of words used in requirements description, in order to increase the requirements specification quality and to be compliant with the client needs.

The structure of this paper is as follows: Section II exposes the proposed model of Requirements Engineering Process, in Section III, we present a Case Study to evaluate and validate the process identifying benefits and considerations about its use. Section IV presents the related work regarding processes that deal with requirements quality assurance and a Comparative Evaluation of the proposed process. Finally, in Section V, we present our Conclusion and Future Work.

II. THE PROPOSED PROCESS

In this section, we present the proposed process model to provide assistance to the *REng* in requirements analysis, specification and validation. The process seeks to increase the description standardization and minimizes the inconsistencies occurrence in requirements.

The process model is divided into three phases: Analysis, Specification and Validation. We also present a description for each phase, showing what is covered in that phase, the goals and its input and output artifacts.

A. Analysis

Input Artifacts: Description of client needs.

Description: **1st Step:** The process begins when the client and the *REng* interact in iterative and incremental meetings, debating the system requirements. These meetings may be held where the client deems necessary; in most cases they happen at the client's company. Every new meeting resumes the issues discussed and the needs already identified in an incremental manner, until a final consensus is established.

2nd Step: Next, the *REng* transcribes the needs, passed by the client, and identified as possible system requirements.

Goals: Previous works show that this is the moment with a strong probability that the information being passed by the client is inaccurate or it does not truly represent the client needs [10][15][18]. So, in this phase, it is extremely important that the *REng* gets as much information as possible about the client needs for the system, so that he can translate these needs into requirements.

Output Artifacts: Description of needs transcribed by the *REng*.

B. Specification

Input Artifacts: Description of needs transcribed by the *REng*.

Description: 1st Step: Having knowledge about client needs, the REng can identify, or create, what we call General Context. These are words that will identify where the specified requirement will be contained. It is used, for example, for a project developed in various modules, as the requirements of each module will be separated from the General Context identifying them, facilitating future search.

The Specific Context has the same function as the General Context, but makes the area where the requirement will be included more specific. Using the same idea from the example above, in a project with several modules, the Specific Context will point within each module, where the requirements being specified will be contained.

2nd Step: After the contexts creation, the REng describes into requirement, using words in natural language, the need already identified by the client.

The process model does not use a treatment for synonymous words. Each new word typed, if it has not previously been validated, is classified as new word and not reused, even if it is a synonymous of a word already validated and stored in RHBD.

In Figure 1, we show an example of the beginning of a requirement description. We are also presenting some examples of suggested words.

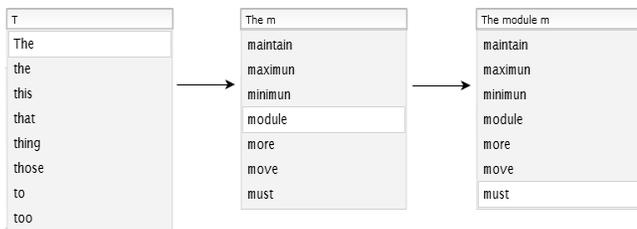


Figure 1: Requirement description example

3rd Step: Following the description, our process has a requirement classification by functionality. According to [17], they are treated as Functional Requirements and Non-Functional Requirements.

The differential of the proposed process occurs during the 1st. and 2nd. steps of this phase. As the words are described, both in General and Specific Context and in the requirement description, we propose an aid to the REng when the requirement is being described. Words suggestions, previously used in another specification, will be offered to the REng, which he can use or not to continue describing the requirement.

It is important to point out that, even the reuse of words proposed by the model being optional to the REng, if the words are not reused this will reflect in the requirement validation. The expected quality will not be achieved, resulting in a negative validation.

For the proposed model, a requirement will be complete only when connected to a General and Specific Context, its description is complete and classified as Functional or Non-Functional Requirement.

Goals: In this phase, the process has four goals:

- Facilitate the requirements separation, using General and Specific Contexts, based in the [2][3][10] works;
- Clarify ambiguities that may occur in a larger project, when some requirements become much like others;
- Description Standardization, since the REng has available words suggestions;
- Reuse of words, differently of the proposed reuse in [11], our process proposes the reuse of already used words, and points the possibility of partial or total requirement reuse.

Output Artifacts: Requirement Specification.

C. Validation

Input Artifacts: Requirement Specification.

Description: 1st Step: According to the rigorousness levels pre-defined by the REng, the description of General and Specific Contexts and the requirement description are evaluated and validated.

The validation occurs together with the client, based on what we call Requirements History Database (RHBD). It contains all the words utilized in any requirements specification in any project within the organization.

When we deal with requirements validation based on RHBD, some care is needed to ensure the new specified requirement, which uses words from others validated requirements, has the desired quality [13]. For this reason, the process model proposes a complete requirement validation (General and Specific Context, Description and Classification), with the client, checking the consistency of all these items.

2nd Step: If REng and client understand the validation as negative, we encourage the change of items (general and specific context and requirement description) that contains inconsistencies. After these changes are made, a new validation may occur. A negative validation means that the client need was not correctly translated in the form of requirement. This may occur because he has passed incorrect information or the REng cannot correctly identify the requirement.

3rd Step: If REng and client understand the validation as positive, we propose that all items are stored. They will be in a list we call Requirements List, containing all the requirements identified, and then stored in the RHBD. A positive validation represent that the client was able to read, understand and validate the requirement specified by the REng.

Goals: Validate if the described requirement reflects the needs described by the client, and if it contains the proposed description standardization and expected quality. At this moment, the client may realize the translation of his need, previously expressed, to a requirement for the system. Thus, his analysis, along with the REng, can confirm that it is actually the desired requirement.

Output Artifacts: Requirements List containing all the specified requirements.

We exemplified, in Figure 2, the process model proposed by this paper, with its phases and input and output artifacts.

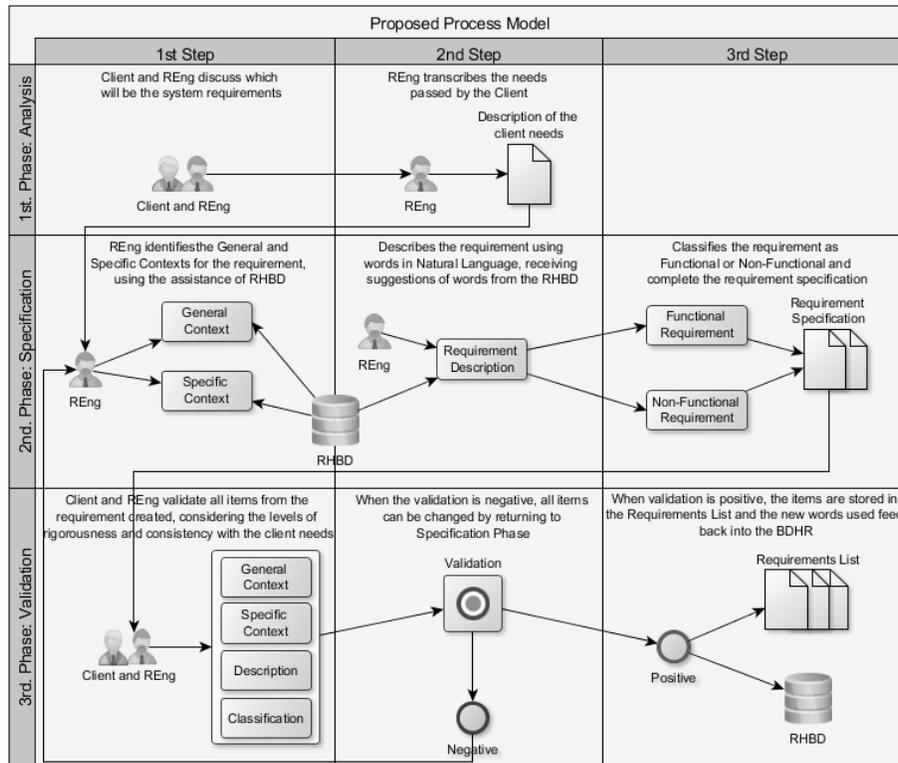


Figure 2: Proposed Process Model

At the end of all meetings, after all requirements are defined and validated, it would be possible to create a Software Requirements Document using the Requirements List the proposed process. Our process does not specify a model or a default template, because it is not the intention of this work. We only seek to offer all requirements previously validated and stored, assigned to a software project under development.

III. CASE STUDY

For this case study, we selected two modules of an academic project of the Software Factory – GAIA. It is located in the Computer Science Department at the State University of Londrina. The modules have been developed in partnership with the University Dental Clinic (UDC), an agency owned by the University. The project was to deploy an electronic health record supporting the activities developed in the Odontology Department.

The objective of this case study is evaluate the quality increase in the requirements description, using the process model proposed, and minimizes the occurrence of non-standardized requirements.

We use the concepts presented in [6][16] to report, quantify and evaluate the results obtained during this study.

The development team was composed of four graduate students in the role of the implementation team, and two master students in the role of the REng. In this study, the role of the Client was played by teachers responsible for the clinical (disciplines) of Pediatric Dentistry and Geriatric Dentistry attended by UDC.

In this case study, the master students playing the role of REng have only academic experience, so the proposed process was applied without the experience of an expert in the software development industry. However, the analysis and results of the case study were consistent and within the expected ranges.

Both modules were located in the same project, but they have been developed separately, such that the results of the requirements specification could be compared. The Pediatric module used the proposed process and the Geriatric module was developed without the use of the proposed process.

A. Data Analysis

For the data analysis, five items were chosen, reflecting the metrics that describe the data for the two modules. Thus, we could evaluate the effectiveness of the proposed process. In Figure 3 we present these metrics.

	Pediatric Module	Geriatric Module
Total of Specified Requirements	35	34
Total of Used Words	853	633
1 Approved by the client (1 st Iteration)	28	20
2 Approved by the client (2 nd Iteration)	35	24
3 Approved by the client (3 rd Iteration)	-	34
4 Description Problems	5	12
5 Context Ambiguity	0	6
6 Number of reused words	777	530
7 Number of non-reused words	76	103

Figure 3: Quantitative Data Analysis

We present in Figure 4, four examples of requirements for the Pediatric module obtained using the proposed

process model, and for the Geriatric module without the use of the model.

Obtained Requirements	Pediatric Module	Geriatric Module
Example 1	The module must provide all fields for the Childhood Model Evaluation with options like "Yes" or "No".	The client wants a field for the patient Local Orientation, so he can tell where he is.
Example 2	The module shall maintain listed the entire patient's history, as the 5th item in the Oral Hygiene Index.	There should be a location on the page to store the history of points obtained by the patient.
Example 3	The module must include at the bottom of the evaluation page two fields, in order to specify the treatment results and patient progress.	On the Geriatric page the dentist will note the score of the Evocation Test, the patient needs to remember the result of a subtraction.
Example 4	The module shall include, at the 4 th item on the page, the Alimentation History, with options to indicate how many times a day the patient feeds on each item.	The system will allow the dentist to evaluate the patient's attention through a test.

Figure 4: Requirements obtained in the Case Study

These examples were chosen to demonstrate the results qualitatively. In addition, they have been specified in the first iteration of each module, i.e., some of them have inconsistencies which will be discussed below.

1) Requirements Approved by the Client

This metric consisted in the analysis of the entire description of the client needs, specification of these needs in the requirement form, and finally, the validation of the requirements created by the REng and client. In this case study, 2 meetings were necessary until all requirements were validated. The 1st. meeting consisted in one iteration and the 2nd. meeting in two iterations, with total of three iterations as indicated in Figure 3.

Analyzing the data presented, in the first iteration the process achieve 80% of requirements approval against 58% without the process. In the second iteration, the proposed process model has reached 100% of requirements approved against 70%, requiring a third iteration, to achieve 100% of requirements approved.

This metric analysis allows us to identify that using the proposed process model has increased by 22% the number of requirements approved in the first iteration and by 29% the number of requirements approved in the second iteration.

Another important fact is the reduction in the number of iterations in the meetings with the client. This reduction was achieved through the reuse of words based on RHBD, and also the experience of the REng. These make it possible to translate the client needs more quickly, increasing the specification quality.

2) Description Problems

Here, we present the number of requirements that have experienced problems in their description. They could be writing errors or requirements that do not consistently represented the need described by the client.

In the requirement of Example 4, Geriatric module, there is an inconsistency which generated a negative validation by the client. The description:

“The system will allow the dentist to evaluate the patient's attention through a test.”

presents an error that was corrected in the 2nd iteration, since the client need was:

“The module must allow the dentist to evaluate and record, through a field, the degree of the patient attention, when asked to recall the name of a shown object.”

In the same Example 4, now in Pediatric module, the requirement was positive validated in the first iteration. The description:

“The module shall include, as the 4th item on the page, the Alimentation History, with options to indicate how many times a day the patient feeds on each item.”

correctly denoted the client need, and have the expected standardization the model proposes.

From the 35 requirements specified in the Pediatric module, 5 requirements were identified with some of the problems cited, while in the Geriatric module, the number of problematic requirements was 12 out of 34 requirements. This data indicates a decrease in the amount of requirements description problems by 21% when using the proposed process model.

3) Context Ambiguity

This metric was used to evaluate the effectiveness of the process model 2nd Phase. With the identification and allocation of General and Specific contexts for the specified requirement, this became distinguishable from other similar requirements, making it clear where in the project the requirement specified is contained.

To demonstrate the results of this metric we used the requirement from Example 2, which shows how General and Specific contexts helped to minimizes the ambiguity occurrence in the requirements.

In Figure 5, we exemplify how the requirement was treated using the ideas proposed by the process model.

General Context	Health Record Management
Specific Context	5th Item Pediatric module
Description	The module shall maintain listed the entire patient's history, as the 5th item in the Oral Hygiene Index.
Classification	Functional Requirement

Figure 5: Use of Requirement Contexts

In comparison, the same Example 2, but now the specified requirement for Geriatric module presents a context ambiguity inconsistency. In the description:

“There should be a location on the page to store the history of points obtained by the patient.”

it is not clear where this requirement is inserted into the project, much less what's the module it belongs to or what would this "location on the page" the client expects. We could easily transport this requirement to another project

module with high consistency possibility within this module as well.

There was no context ambiguity in the requirements specified using the proposed process model. However, there were 6 inconsistencies with context ambiguity in the requirements specified in the Geriatric module. This data allows us to identify an improvement by 17% concerning problems of context ambiguity of the specified requirements.

4) *Number of reused words*

The number of reused words in the requirements description was also chosen as a metric, for evaluating the effectiveness of the proposed process model.

All words that were reused at least once in the requirements specification were identified in this analysis. There were also words initially contained in RHBD, so if these words were used they also are identified as reused.

Another important item is the initial word amount contained in RHBD. For both modules, there were 187 available words in RHBD. As the requirements of Pediatric module were specified using the proposed process model these words were used in 2nd. Phase of the process. In the Geriatric module, which did not use the process, we verified if these words were used.

In the Pediatric module 853 words total were used to specify 35 requirements; 777 words were reused, generating a reuse percentage of 91%. In the Geriatric module, 633 words were used to specify 34 requirements; out of these, 530 words were reused, generating a percentage of 83% reuse. Through these data, we can confirm an increase of 7%, by using the proposed process model.

5) *Number of Non-Reused Words*

We also use a metric to evaluate the number of non-reused words identified. Thus, we could confirm how the reuse of words affects the requirements description.

To calculate this metric, all words used in the requirements specification have been checked, and if the word was used only once it was identified as not reused.

In the Pediatric module, 76 words were used only once, making 8.9% of the total words. In the Geriatric module, 103 were used only once, making 16% of the total. These data shows a reduction of 7% in non-reused words.

The results of the last metrics are shown in Figure 6.

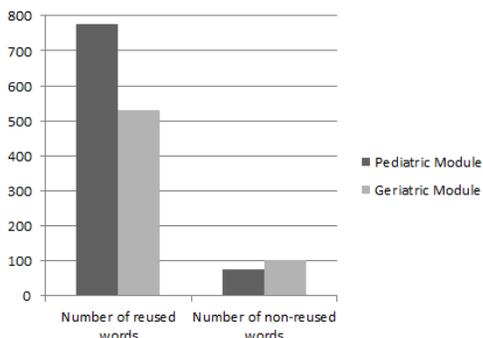


Figure 6: Analysis of Reused and Non-Reused Words

They demonstrate that the relationship is equal to an inverse proportionality, with 7.4% increase in the reuse and 7.3% decrease in non-reuse.

Finally, we emphasize that, this case study objective was to evaluate the quality increase in the requirements description by using the proposed process model, reducing the occurrence of non-standardized requirements.

The requirements specified using the proposed model had a better description, because they were based on words already used and validated by the client, allowing its standardization. The requirements specified without the help of the model were dependent on the REng knowledge and experience, confirming what had already been cited in [14][15][18].

IV. RELATED WORK

In the literature, there are several scientific papers and books dealing with the requirements quality assurance. They point out benefits and provide experiences that help us better understand how to create and maintain a requirements specification with quality.

Thus, through a literature review, three processes dealing with requirements quality assurance were selected. Others could be chosen, which directly address the phases of requirements engineering process, but the purpose of this section is to present processes using divisions in phases, contexts and perspectives in order to improve the requirements treatment receive during their specification.

In their work, Chen et al. [2] described a technique that uses a pre-processing of natural language in software requirements creation. This pre-processing makes use of general and specific fields to separate the requirements. After this, the technique does a search for words, called "objective" by the author, which are described as the central part of the requirement.

According to Cabral et al. [1], the application of systematic reading techniques such as Perspective-Based Reading (PBR) and nonsystematic as Checklist during the requirements analysis has brought good results. In these techniques, several inspectors inspect a software context document looking for errors or inconsistencies before transcribing the requirements document. These errors are then evaluated to compare the two techniques.

A model of Requirements Engineering Process has been proposed in the work of Pandey et al. [12]. The authors cover the entire area of requirements engineering, proposing the division into four phases: Requirements Elicitation and Development, Requirements Documentation, Requirements Verification and Validation and lastly Requirements Planning and Management. The requirements are stored in a Software Requirement Specification (SRS) and the authors point out that the differential of their work is, besides covering all areas, enabling the Changing Management in requirements already agreed.

A. *Comparative Evaluation*

This subsection presents a comparative evaluation of the proposed process model and the works discussed. Figure 7 shows this comparison between processes.

It is also important to note that the comparison made sought papers that deal with quality assurance requirements through processes and techniques to achieve a better quality specification.

With this comparison, we want to reinforce the ideas already identified in [8][9], and served as base for creating the process model presented in this paper. These ideas

concentrate on the use of specific areas to treat and group requirements, using explicit contexts to prevent inconsistencies and reduce redundancies in the reuse of conflicting requirements. They also present a writing standardization through the reuse of words that will form the requirements.

	Chen et al.	Cabral et al.	Pandey et al.	Proposed Process
Requirements Identification	Client creates the requirements.	Not specified in the work.	Raw requirements are identified by the client's views.	Client and REng identify the requirements during the meetings.
Requirements Analysis	There is no analysis of the requirements.	Not specified in the work.	A System Analyst analyzes the raw requirements and creates the system requirements.	REng does the requirements analysis based on his experience and contexts.
Requirements Specification	Specification comes from the identification of requirements made by the client, written in NL.	Requirements are already specified, contained in the Requirements Document.	Specification is made through the SRS.	Specification is made by the REng, using NL, with the option to choose words already validated.
Requirements Validation	It is not the objective of the work validate the requirements.	Focuses on the requirements validation using reading techniques and requirements review.	Validation is done by analyzing the SRS along with the client.	Requirements are validated by client and REng, through levels of rigorosness.
Uses group of requirements	Yes, using General and Specific Domains.	No	Yes, in levels of the system.	Yes, using two levels: Contexts and Types.
Uses standardized requirements	No	No	No	Yes
Requirements Quality	Increase the quality of the very requirements.	Increase the quality of the requirements validation.	Increase the quality of the whole process.	Increase the quality of the requirements validation and description.
Level of Client Participation	High, identifying and specifying the requirements.	The work does not cite how the client participation was.	Client participation must be constant.	High, identifying and validating the requirements.
Level of manual interaction	Medium, the model uses computational assistance for the treatment of requirements.	High, the model uses techniques that require a lot of human interpretation.	High, the model does not mention possibilities of process automation.	Medium, the model presents the possibility of automation for the requirements handling.
Partial or Total Reuse of Requirements	No	No	No	Yes

Figure 7: Comparison between Processes

This analysis helps us realize that the requirements description standardization, as well as their total or partial reuse are significant issues in the search for quality increase.

Even when studies address requirements quality assurance in a software specification, these items have a very favorable area for research and development.

The proposed process model differs from the works presented by the following:

- In the requirements specification, when it is proposed the suggestion of validated words for the requirements description;
- In the requirements validation, with client and the REng validating the requirements, using levels rigorousness, so the words used and also the requirements created can be reused;
- In the grouping requirements, offering the division into general and specific contexts and also functionality;
- In the description standardization, through the reuse of words, always seeking to achieve a requirements quality increase.

V. CONCLUSION AND FUTURE WORK

The description, documentation, and requirements reuse were the main focus of the paper presented. Also, problems encountered in the requirements description, non-standardization and inconsistencies with client needs, were some of the issues that we tried to expose and propose a way to improve.

The proposed process model improves the requirements written standardization and minimizes the chance of description inconsistencies.

The case study showed that it is possible to obtain quantitative and qualitative improvements at the time of specification, corresponding to 22%; in the reduction of requirements description and ambiguity problems, with gains of 21% and 17% respectively; and finally allowing an increase of 7% in the reuse of words that compose the requirements.

As future work, we seek to improve the requirements documentation proposed by the process model, provide an alignment between the proposed process and the Requirements Management, so that, in addition to being treated, requirements can also be managed.

The case study presented allowed the evaluation of the process model using two different modules, which is comparable with a software development. Our next step is to apply the process model in a same module or software, and evaluate the results, obtaining even more data to support the process model effectiveness.

We also seek to develop and use a CASE tool that implements all concepts presented in the process, so we can use computer assistance and further increase the benefits obtained. In addition, we can evaluate the performance of the RHBD when operating with a very large number of words.

REFERENCES

- [1] M. S. Cabral, F. Alencar, J. Castro, O. Pastor, J. Sanches, "Aplicação de técnicas de leitura durante a análise de requisitos", WER08 - Workshop em Engenharia de Requisitos, pp. 193-204, Barcelona, Catalonia, Spain, 12-13 September 2008.
- [2] H. Chen, H. Keqing, P. Liang, R. Li, "Text-based requirements pre-processing using nature language processing techniques". International Conference on Computer Design and Applications (ICCCA), pp. 14-18, Qinhuangdao, Hebei, China, 25-27 June 2010.
- [3] J. Cybulsky, and K. Reed, "Requirements classification and reuse: crossing domains boundaries". 6th International Conference on Software Reuse, pp. 190-210, Viena, Italy, 2000.
- [4] IEEE 830: Recommended Practice for Software Requirements Specification, http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=15571&arnumber=720574&count=1&index=0, retrieved: July, 2012, 1998 (R2009).
- [5] IEEE 1233a: IEEE Guide for Developing System Requirements Specifications, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=741940&contentType=Standards>, retrieved: July, 2012, 1998.
- [6] B. Kitchenham, et al.: Evaluating guidelines for reporting empirical software engineering studies. *Empirical Software Engineering* 13 (1), October 2007, pp. 97-121, doi:10.1007/s10664-007-9053-5.
- [7] A. V. Knethen, B. Paech, F. Kiedaisch, F. Houdek, "Systematic requirements recycling through abstraction and traceability". *RE - Requirements Engineering*, pp. 273-281, Essen, Germany, 2002.
- [8] G. Kotonya, and I. Sommerville, *Requirements Engineering: Processes and Techniques*. 1 ed. Wiley, 1998.
- [9] W. Lam, T. A. McDermid, A. J. Vickers, "Ten steps towards systematic requirements reuse". Third IEEE International Symposium on Requirements Engineering, pp. 6-15, 1997.
- [10] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. 1 ed. Wiley, 2009.
- [11] B. Moros, C. Vicente-Chicote, A. Toval, "Metamodeling variability to enable requirements reuse". *EMMSAD - Exploring Modeling Methods for Systems Analysis and Design*, pp. 140-154, Montpellier, France, 16-17 June 2008.
- [12] D. Pandey, A. K. Ramani, U. Suman, "An effective requirement engineering process model for software development and requirements management". International Conference on Advances in Recent Technologies in Communication and Computing, IEEE Press, pp. 287-291, 2010.
- [13] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*. 1 ed. Springer, 2010.
- [14] R. S. Pressman, *Software Engineering - A Practitioner's Approach*, 7 ed. McGraw-Hill, 2011.
- [15] S. Robertson, and J. Robertson, *Mastering the Requirements Process*. 2 ed. Addison Wesley, 2006.
- [16] P. Runeson, and M. Höst, Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14 (2), April 2009, pp. 131-164. doi=10.1007/s10664-008-9102-8.
- [17] I. Sommerville, *Software Engineering*, 8 ed. Addison Wesley, 2007.
- [18] I. Sommerville, and P. Sawyer, *Requirements Engineering: A Good Practice Guide*. 1 ed. Wiley, 1997.