# A MODEL OF REQUIREMENTS ENGINEERING PROCESS FOR STANDARDIZATION AND QUALITY INCREASE

Jose Andre Dorigan and Rodolfo Miranda de Barros
*Computer Science Department - State University of Londrina - Londrina, PR, Brazil*

**ABSTRACT**

One of the biggest problems encountered in Requirements Engineering is the fact that requirements are poorly specified, inconsistent with the client needs or badly written. This paper presents a model of Requirements Engineering Process for requirements description standardization, through the reuse of words, seeking to improve the specification quality. The proposed process will act in Requirements Specification assisting the Requirements Engineer on writing requirements in Natural Language. We present a Case Study to evaluate and identify benefits of its use in a software development.

**KEYWORDS**

Requirements Reuse, Requirements Standardization, Quality on Requirements Description.

## 1. INTRODUCTION

To raise requirements correctly is one of the most important tasks in software development. Requirements are linked to the main problems of software development, in most cases they do not reflect the real needs of users, because they are incomplete or inconsistent (Pressman, 2011; Sommerville, 2007).

A major difficulty is to make the requirements specification be in accordance with the client ideas (Sommerville, 1997). Most often happens a misinterpretation by the Requirements Engineer (we will use the abbreviation *REng* in this work) or the client cannot clearly express their real needs (Robertson, 2006). Anyway, these specification problems create **non-standardized and inconsistent requirements**.

The process model proposed in this paper is justified on the idea that when the requirements description standardization is done through the reuse of words, it's identified 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 (Kotonya, 1998).

The structure of this paper is as follows: Section 2 presents the related work regarding processes that deal with requirements quality assurance; in Section 3 is exposed the proposed model of Requirements Engineering Process. Section 4 shows a Case Study to evaluate and validate the process identifying benefits and considerations about its use. Finally, in Section 5 we present our Conclusion and Future Work.

## 2. RELATED WORK

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, 2010) 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, 2008) 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.

A model of Requirements Engineering Process has been proposed in (Pandey, 2010). 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.

This analysis helps us realize that the requirements description standardization, as well as their 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. Thus, this paper addresses these two items with more importance, to achieve the increase cited.

# 3. THE PROPOSED PROCESS

In this section we present the proposed process model to provide assistance to the REng. 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.

## 3.1 Analysis

**Input Artifacts**: Description of client needs.

**Description**: **1st. Step**: The process begins when client and REng interact in iterative and incremental meetings, **debating which will be 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 the issues discussed and the needs already identified, are resumed to be incremented until a final consensus is established.

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

**Goals**: The experience and works of (Robertson, 2006; Lamsweerde, 2009) show that in this moment there is a strong possibility of information being passed by the client are inaccurate, or not truly represent the need he has (Renault, 2009).

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

## 3.2 Specification

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

**Description**: **1st. Step**: Having knowledge about client needs, the REng can identify, or create, what we call **General Context**, which are words that will identify where the specified requirement will be contained, and the **Specific Context** that has the same function as the General Context, but becoming the area where the requirement will be included more specific.

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

**3rd. Step**: Following the description, the process has a requirement classification by functionality, treating it as **Functional Requirements** and **Non-Functional Requirements** (Sommerville, 2007).

It's 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.

**Goals**: In this phase our process has four goals: **Facilitate the requirements separation**, using General and Specific Context, based in (Chen, 2010; Lam, 1997; Cybulsky, 2000), **Clarify the redundancies** that may occur in a larger project when the requirements become much like others. **Description Standardization** and the **Reuse of words** that is different from the proposed reuse in (Moros, 2008), our process proposes the reuse of already used words and points the possibility of partial or total requirement reuse.

**Output Artifacts**: Requirement Specification.

## 3.3 Validation

**Input Artifacts**: Requirement Specification.

**Description**: **1st. Step**: According to rigorousness levels pre-defined by the REng, the General and Specific Context and the requirement description are evaluated and validated. The validation occurs together with the client, checking if they were used properly and are representing the expected need, based on what we call **Requirements History Database (RHBD)** that contains all the words utilized in any requirements specification for projects within the organization.

**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. A negative validation can represent that the client need was not correctly translated in the form of requirement

**3rd. Step**: If REng and client understand the validation as **positive**, we propose that all items are stored first in a common list we call Requirements List, 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.

We exemplified in Figure 1 the process model proposed by this paper.

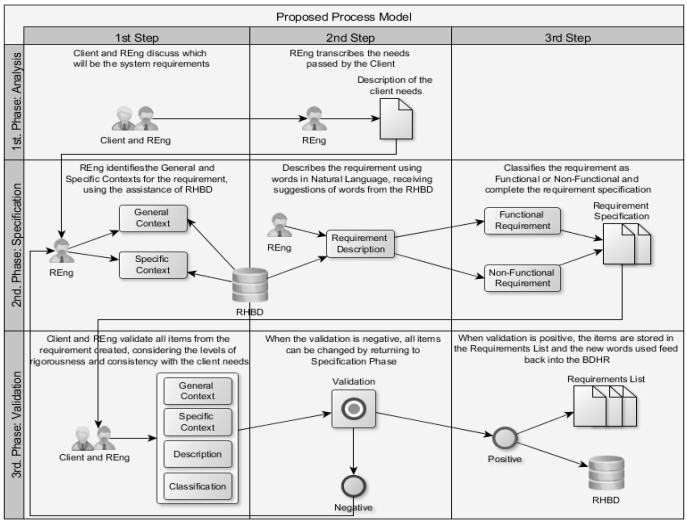


Figure 1. Proposed Process Model

**Goals**: Evaluate whether the described requirement is within the expected standards, for both REng and client. In this moment the client may realize the translation of his need, previously expressed.

**Output Artifacts**: Requirements List containing all the specified requirements.

After the end of all meetings, and all requirements defined and validated, would be possible to create a Software Requirements Document using the Requirements List the process proposed.

# 4. CASE STUDY

For this case study we selected two modules of an academic project developed in a Software Factory located in a Computer Science Department with partnership of the University Dental Clinic (UDC), in order to deploy an electronic health record supporting the activities developed in the Odontology Department.

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

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

## 4.1 Data Analysis

In the data analysis five items were chosen, reflecting the metrics that describe the data for the two modules and evaluating the effectiveness of the proposed process. In Figure 2 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 (1st Iteration) | 28 | 20 |
| | Approved by the client (2nd Iteration) | 35 | 24 |
| | Approved by the client (3rd Iteration) | - | 34 |
| 2 | Description Problems | 5 | 12 |
| 3 | Context Ambiguity | 0 | 6 |
| 4 | Number of reused words | 777 | 530 |
| 5 | Number of non-reused words | 76 | 103 |

Figure 2. Quantitative Data Analysis

**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.

Analyzing the data presented, in the first iteration the process achieve 80% of requirements approval against 58.8% without the process. In the second iteration, the proposed process model has reached 100% of requirements approved against 70.5%, requiring a third iteration, to achieve 100% of requirements approved.

This metric analysis allows us to identify that using the process model proposed has increased in 22.2% the number of requirements approved in the first iteration and 29.5% in the second iteration.

**2. Description Problems**: Here we present the number of requirements that have experienced problems in their description, being writing errors or requirements that do not consistently represented the need described by the client. From the 35 requirements specified in the Pediatric module 5 requirements were identified with some of the problems cited, while in the Geriatric module were identified 12 out of 34 requirements. These data indicates a decrease in the amount of requirements description problems in 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.

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. These data identify an improvement of 17.6% regarding problems of context ambiguity.

**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. Another important item to note is that there were 187 initial words contained in RHBD for both modules.

In Pediatric module were used a total of 853 words to specify 35 requirements, being reused 777 words, generating a reuse of 91.1%. In Geriatric module 633 words were used to specify 34 requirements, of these 530 words were reused, getting a reuse of 83.7%. Through these data we can confirm an increase of 7.4%, by using the proposed process model, in the reuse of words that compose requirements.

**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. In the Pediatric module there were 76 non-reused words, corresponding to 8.9%. In the Geriatric module there were 103 non-reused words, making a 16.2%. These data show a reduction of 7.3% in the non-reused words.

The results of these two last metrics are shown in Figure 3, demonstrating 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 wish to emphasize that this study case 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, allowing its standardization, while the requirements specified without the help of the model were dependent on the REng knowledge and experience, confirming what had already been cited in (Pressman, 2011; Sommerville, 1997; Robertson, 2006).

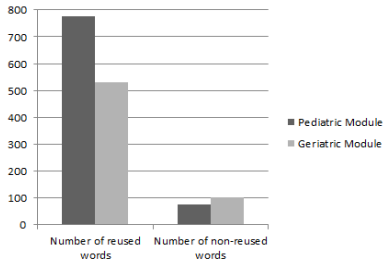


Figure 3. Analysis of Reuse and Non-Reused Words

## 5. CONCLUSION AND FUTURE WORK

This paper showed the effort that exists to make description, documentation, and requirements reuse with a quality higher level, using processes for client and REng can be sure that the requirement created exactly matches the requirement desired.

The case study has revealed that it's possible to obtain improvements at the specification time, corresponding of 22%, reduction of requirements description and ambiguity problems in 21% and 17% respectively and finally allowing an increase of 7.4% in the reuse of words that compose the requirements.

As future work we seek to improve the requirements documentation proposed by the process, provide an alignment between the proposed process and the Requirements Management, so besides treated, requirements can be managed. 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.

## REFERENCES

Cabral, M. S., et al, 2008. Aplicação de Técnicas de Leitura durante a Análise de Requisitos, *WER08 - Workshop em Engenharia de Requisitos*, Barcelona, Catalonia, Spain, 12-13 September, pp 193-204.

Chen, H., et al, 2010. Text-based requirements pre-processing using nature language processing techniques. *International Conference on Computer Design and Applications (ICCDA)*, Qinhuangdao, Hebei, China, 25-27 June, pp 14-18.

Cybulsky, J., Reed, K., 2000. Requirements Classification and Reuse: Crossing Domains Boundaries. *6th International Conference on Software Reuse*, Viena, Italy, pp 190-210.

Kotonya G., Sommerville, I., 1998. *Requirements Engineering: Processes and Techniques*. 1 ed. Wiley.

Lam, W., McDermid, T. A., Vickers, A. J., 1997. Ten steps towards systematic requirements reuse. *Third IEEE International Symposium on Requirements Engineering*, pp 6-15.

Lamsweerde, A. van, 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. 1 ed. Wiley.

Moros, B., Vicente-Chicote, C., Toval, A., 2008. Metamodeling Variability to Enable Requirements Reuse. *EMMSAD - Exploring Modeling Methods for Systems Analysis and Design*. Montpellier, France, 16-17 June.

Pandey, D., Ramani, A. K., Suman, U., 2010. 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.

Pressman, R. S., 2011. *Software Engineering - A Practitioner's Approach*, 7 ed. McGraw-Hill.

Renault, S., Bonilla, O. M., Frach, X., 2009. PABRE: Pattern-Based Requirements Elicitation. *Third International Conference On Research Challenges in Information Science*, Fez, Morocco, April 22-24.

Robertson, S., Robertson, J., 2006. *Mastering the Requirements Process*. 2 ed. Addison Wesley.

Sommerville, I., 2007. *Software Engineering*, 8 ed. Addison Wesley.

Sommerville, I., Sawyer P., 1997. *Requirements Engineering: A Good Practice Guide*. 1 ed. Wiley.