DoMAR: An APPROACH TO PREVENT PROBLEMS RELATED TO REQUIREMENTS DOCUMENTATION AND MANAGEMENT

Data de entrega dos originais à redação em:04/02/2016

e recebido para diagramação em:30/11/2016

Milene Elizabeth Rigolin Ferreira Lopes Salvador[1]
Luciana Brasil Rebelo dos Santos[2]

The suitable accomplishment of the Requirements Engineering (RE) process in the early stages of software engineering project is one of the safest ways to achieve quality in the development of the software product. However, despite the growing number of projects which use the RE process, experts realize that the process is not well structured and executed, leading to many problems, both in collecting and in the documentation and management of requirements. One approach to assist in structuring and implementing the RE process is to provide ways to prevent some of the problems. Based on this assumption, the present work presents DoMaR (Documentation and Management Requirements), an approach that maps the problems in requirements documentation and management, in order to propose a solution to prevent such detected problems. An environment, which has the same name of the approach, was developed to assist in requirements registration, organization, and monitoring. A case study using the proposed environment as well as the results of applying the approach is presented.

Keywords: Requirements Engineering. RE Problems. Documentation. Management. Traceability.

A realização adequada do processo de Engenharia de Requisitos (RE) nas fases iniciais do projeto de engenharia de software é uma das formas mais seguras de alcançar qualidade no desenvolvimento do produto. No entanto, apesar do número crescente de projetos que utilizam o processo de RE, os especialistas percebem que este processo não está bem estruturado e executado, levando a muitos problemas, tanto na coleta quanto na documentação e gerenciamento de requisitos. Uma abordagem para auxiliar na estruturação e implementação do processo de RE é fornecer maneiras de evitar alguns dos problemas. Com base neste pressuposto, o presente trabalho apresenta DoMaR (Documentação e Requisitos de Gestão), uma abordagem que mapeia os problemas na documentação de requisitos e na gestão, a fim de propor uma solução para prevenir os problemas detectados. Um ambiente, que tem o mesmo nome da abordagem, foi desenvolvido para auxiliar no registro de requisitos, organização e monitoramento. É apresentado um estudo de caso utilizando o ambiente proposto, bem como os resultados da aplicação da abordagem.

Palavras-chave: Engenharia de Requisitos. Problemas RE. Documentação. Gestão. Rastreabilidade.

[1] Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Caraguatatuba, milenerigolin@ifsp.edu.br
[2] Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Caraguatatuba, lurebelo@ifsp.edu.br

1 INTRODUCTION

The main problems relating to a software project development using traditional methodology are associated to the Requirements Engineering (RE) area. These problems are due to the fact that in the RE most activities are carried out manually, and thus are more liable to human error. One of the main consequences of poorly specified requirements is rework, which can consume 30% to 50% of the total development cost (BOEHM; PAPACCIO, 1988 cited WIEGERS, 2003). Furthermore, errors in requirements cause 70% to 85% of the cost of rework. (LEFFINGWELL, 1997 cited WIEGERS, 2003).

Considering that, this paper presents an environment, called DoMaR (Documentation and Management Requirements), to assist in requirements registration, organization, and monitoring. DoMaR enables the Requirements Engineer to perform the management and documentation in an environment that is designed to prevent some common problems when dealing with requirements, such as: requirements loss, absence of requirements traceability, lack of requirements prioritization, and discard of important information.

The methodology used for the research development consists of: (1) Literature review in RE context; (2) Survey of problems in management and documentation requirements; (3) Development of an environment prototype that supports the RE process; (4) A mapping between detected problems and the resources / functions in the environment developed; (5) Applying the environment in a case study.

2 REQUIREMENTS ENGINEERING

The RE is a sub-area of Software Engineering that provides methods, techniques, and tools that assist the process of requirements collection, documentation, and management (SOMMERVILLE, 2007). A requirement can be defined as descriptions of services provided by the system and its operational constraints (SOMMERVILLE, 2007).

The IEEE Standard Glossary of Software Engineering Terminology (1990) defines requirement as: (1) a necessary condition or capacity to be provided by the system under development, so that a user can solve a problem or achieve a goal; (2) A condition or capacity to be found in the system or a system component to satisfy a contract, standard or other document formally instituted; and (3) a documented representation of a condition or capability as in items 1 or 2 (ELECTRICAL; ENGINEERS, 1990).

According to WIEGERS (2003), requirement is a property that the product must contain in order to provide to stakeholders. YOUNG (2004) defines requirement as a necessary attribute in a system, a statement that identifies a capacity, characteristic or quality factor, in order to the system be valuable and usefulness to stakeholders. According to SOMMERVILLE (2007), there are two types of requirements:

- **User requirements:** user needs statements and constraints in natural language. It is possible use diagrams to facilitate understanding.
- **System Requirements:** define functions, services, and operational constraints of the system in detail. It is possible use technical and formal language. They are divided into:
  - **Functional requirements:** they describe the system features and how the system should react to specific entries, and behave in certain situations.
  - **Non-functional requirements:** they describe the services restrictions or functions offered by the system.

   - **Domain requirements:** they reflect the application domain characteristics and constraints of the developed system.

The RE process consists of discover, analyze, document, and verify these functions and restrictions in order to generate a document containing the system requirements specification.
It comprises the steps of requirements extraction, requirements analysis and negotiation, requirements documentation, and requirements validation. To organize these steps, there is the area of Requirements Management. These steps do not happen in a strict sequence; they are repeated throughout the development process every time it is necessary (SOMMERVILLE, 2007).

The problems identified in the survey were reported by the authors ALEXANDER AND STEVENS (2002), FIRESMITH (2003), FIRESMITH (2007), WIEGERS (2003), VERNER et al. (2006), and HULL, JACKSON AND DICK (2005).

Following, the detected problems relating to requirements documentation are presented:

- **Confusion between restrictions and requirements** - occurs when there is a misunderstanding of definitions and differences between restrictions and requirements, not being possible to differentiate them.
- **Metadata lack** - consists of lack of descriptive information about a requirement to inform the status and priority of each requirement. Not considering this item brings on outdated requirements and affects the process development organization, causing confusion, duplicated work (two people working in the same requirement), wasted time, among others.
- **Identification lack** – do not identify the requirements by means of codes, numbers, among others.
- **Non-documented requirements sources** - consists of not storing information of requirement source, for example, who requested, from which document it was extracted, and so on.
- **Loss of stakeholders information** – consists of not write down, forget, dismiss information or documents that stakeholders have provided for the requirements basis.
- **Requirements loss** - miss the requirements, caused by deleting the consolidated requirement and having no ways to retrieve it or loss of information.
- **Duplicate information** - consists of statement or writing the same information twice, in the same document.
- **Outdated** - not update the requirements when changes occur.

Following, the detected problems relating to the management requirements are presented:

- **Not tracked requirements** – consists of not storage important information for requirements, such as the sources of requirements. Also consists of not allocating the requirements in their design elements, architecture and test sets.
- **Prioritization lack** - not perform requirements prioritization.
- **Lack of requirements classification** – do not sort the requirements in functional, non-functional, or domain requirements.
- **Erroneously information discard** - delete important information relating to the requirement.
- **Poorly organized requirements** - consists of lack of requirements organization in categories, modules, prioritized, among others. It is also related with the lack of a proper environment for the management of such requirements.

- **Poorly managed requirements** – consists of lack of documentation, historical changes, progress reports of the requirement state. This management lack leads to rework, delays deadlines, among other disorders.
- **Loss of supply requirements** – do not document who requested the requirement or from where it was extracted.

3 DEVELOPMENT

After analyzing the problems relating to requirements management and documentation, it was defined what issues would be addressed in the proposed environment, which are: **Metadata Lack**, **Identification Lack**, **Non-documented requirements sources**, **Requirements loss,** Not tracked requirements, Prioritization lack, **Lack of requirements classification, Erroneously information discard, Poorly organized requirements, Poorly managed requirements, and Loss of supply requirements.**

The other problems were not covered because some of them can not be solved using computing resources, such as: Confusion between restrictions and requirements, Loss of stakeholders information, and Outdated. The issue Duplicate information, although possible, is out of our scope because it is very specific. So, it was decided to keep it for future work. The issue Erroneously information discard is partially treated with the history of modifications. However, it is necessary many other items to cover this item, which were left for future work.

Considering what has been explained so far, the following topics were developed: (1) an environment prototype and (2) a mapping between the identified problems and the resources / functions used in the prototype to prevent these problems.

3.1 DoMaR Environment Prototype

The proposed prototype is an environment to assist in registration, organization, and monitoring of the requirements. This environment has the following features: (1) register: customer, employee, origin and requirement; (2) information change: customer, employee, origin and requirement; (3) requirement tracking and tracking display; (4) requirements reports by: customer, employee responsible, source, category, type, status and priority; (5) requirements changes history.

The interaction between the user and DoMaR can be seen in Figure 1.

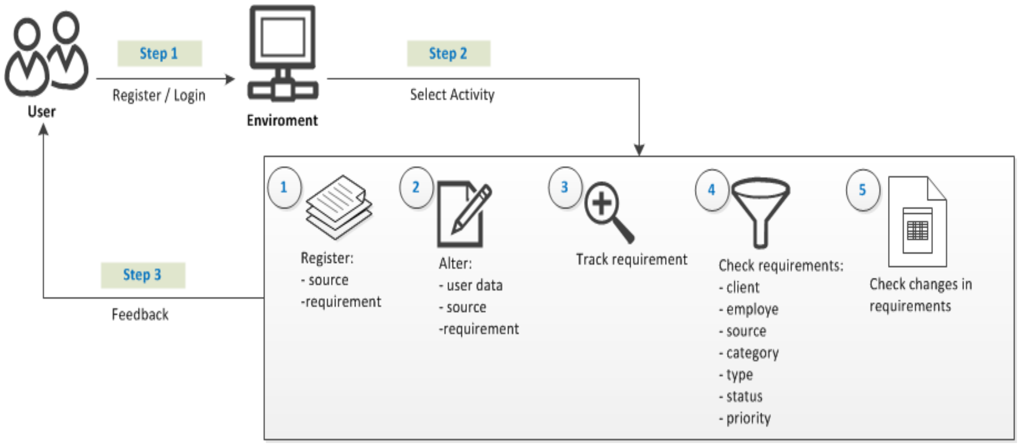
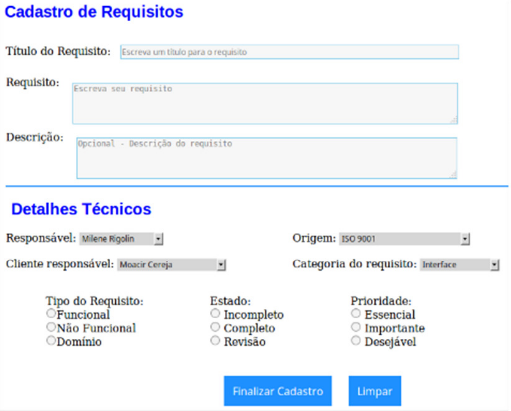


Figure 1 - Interaction between the user and the environment.

Figures 2, 3, and 4 presents screens of DoMaR environment. In Figure 2 it is possible to visualize the requirements register screen. Figure 3 shows the screen of selection and modification of customer data.



Cadastro de Requisitos

Título do Requisito: Escreva um título para o requisito

Requisito: Escreva seu requisito

Descrição: Opcional - Descrição do requisito

Detalhes Técnicos

Responsável: Milene Rigolin

Origem: ISO 9001

Cliente responsável: Moacir Cereja

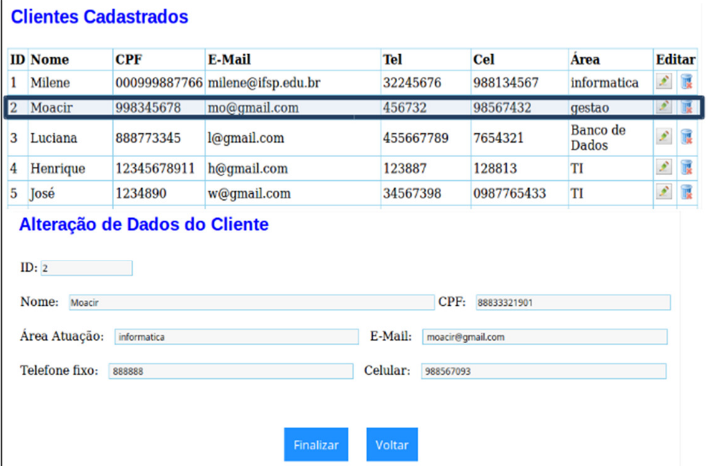
Categoria do requisito: Interface

Tipo do Requisito:
○ Funcional
○ Não Funcional
○ Domínio

Estado:
○ Incompleto
○ Completo
○ Revisão

Prioridade:
○ Essencial
○ Importante
○ Desejável

Figure 2 - Requirements register screen



Clientes Cadastrados

ID	Nome	CPF	E-Mail	Tel	Cel	Área	Editar
1	Milene	000999887766	milene@ifsp.edu.br	32245676	988134567	informatica	
2	Moacir	998345678	mo@gmail.com	456732	98567432	gestao	
3	Luciana	888773345	l@gmail.com	455667789	7654321	Banco de Dados	
4	Henrique	12345678911	h@gmail.com	123887	128813	TI	
5	José	1234890	w@gmail.com	34567398	0987765433	TI	

Alteração de Dados do Cliente

ID: 2

Nome: Moacir

CPF: 88833321901

Área Atuação: informatica

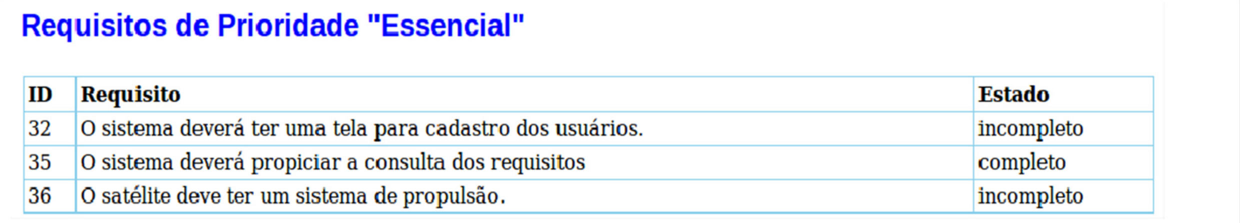
E-Mail: moacir@gmail.com

Telefone fixo: 888888

Celular: 988567093

Figure 3 - Screen of selection and modification of customer data.

Figure 4 presents the screen with the results of a query for organizing the requirements.



Requisitos de Prioridade "Essencial"

ID	Requisito	Estado
32	O sistema deverá ter uma tela para cadastro dos usuários.	incompleto
35	O sistema deverá propiciar a consulta dos requisitos	completo
36	O satélite deve ter um sistema de propulsão.	incompleto

Figure 4 - Screen showing the results of a query for organizing the requirements

3.2. Mapping between the problems and the environment resources / functions

This section presents the mapping carried out between the problems that are addressed in the research and the resources / functions, which were designed in order to prevent such problems.

3.2.1. Problems related to requirements documentation

Metadata Lack: this problem was treated by inserting the status fields, priority and responsible for the requirement when carrying out the registration requirement.

Identification Lack: this problem was treated by inserting a unique number that is generated automatically by the database when recording the requirements.

Non-documented requirements sources: in order to address this problem, fields were created for entering the font name when the requirement is removed as well as the field to mark the customer responsible for the condition.

Requirements loss: This issue was treated by creating requirements log which stores the requirements changes.

3.2.2. Problems related to requirements management

Not tracked requirements: a page was created for performing tracking between requirements as well as the selection fields of the origin and category of the requirement.

Prioritization lack: aiming to preventing this problem, a field was created to select the priority of the registered requirement.

Lack of requirements classification: a field was created to enable marking the type of requirement in order to prevent this problem.

Poorly organized requirements: categories were developed to improve the requirements organization. Filters are also available to view the requirements for each category.

Poorly managed requirements: in order to prevent this problem, a filter was created to view the requirement for state as well as a page with the requirement modifications history.

Loss of supply requirements: to prevent this problem, fields were developed for documenting the sources of the requirements, including the document source as well as the customer responsible for the requirement.

4 CASE ESTUDY

The case study consists of applying the environment in one scenario that simulates the everyday activities in management and documentation of requirements. Therefore, eleven experts were selected to experience the environment. The experts work in real software projects and have experience with requirements.

The main objective of this experiment is to check if the functionalities implemented in environment can prevent the detected problems. The case study preparation consists of the following activities: **(1)** Organization of the case study; **(2)** Questionnaire preparation; **(3)** Expert profile definition; **(4)** Selection of experts; and **(5)** Test execution.

The case study was performed in eight steps: **(1)** Registration and log in; **(2)** Register of four requirements; **(3)** Change the text of a requirement; **(4)** Map the requirements; **(5)** View mapping; **(6)** Explore search filters; **(7)** View change log in requirements; **(8)** Evaluate features of the system.

4.1. Results

The results obtained in the test implementation can be observed in Table 1. In order to analyze and evaluate of the environment functionalities, it was used the Likert scale (LIKERT, 1932), with four factors: 4- very high; 3- high; 2- low; 1- very low.

Table 1 - Case study results

	Problems related to documentation requirements				Problems related to management requirements					
Specialists / Problems	Metadata Lack	Identification Lack	Non-documented requirements sources	Requirements loss	Not tracked requirements	Prioritization lack	Lack of requirements classification	Poorly organized requirements	Poorly managed requirements	Loss of supply requirements
Specialist 1	3	4	4	4	3	4	4	4	4	4
Specialist 2	4	4	4	3	3	2	4	3	4	4
Specialist 3	3	4	4	3	4	3	4	3	3	4
Specialist 4	3	4	4	3	4	3	4	4	3	4
Specialist 5	4	3	4	4	4	4	4	4	4	2
Specialist 6	4	4	4	4	4	4	4	4	4	4
Specialist 7	2	4	2	2	3	3	4	4	3	1
Specialist 8	4	3	4	4	3	4	4	2	4	4
Specialist 9	3	4	4	4	3	4	4	4	4	4
Specialist 10	3	4	4	4	3	4	4	4	3	4
Specialist 11	4	4	4	4	4	4	4	4	4	4
Average	**3,4**	**3,8**	**3,8**	**3,5**	**3,5**	**3,5**	**4,0**	**3,6**	**3,6**	**3,5**

Based on the obtained results, it is possible to conclude that, in the avarage, the functionalities implemented in the proposed environment achieve the objective, wich was to prevent the detected problems. One can also note that the functionalities which obtained results below 3,5 should be improved.

4.1.1. Suggestions given by Specialists

The specialists suggested changes to improve the environment. The suggestions were classified into categories. Relating to historical changes, the suggestion is to visualize the changes in the technical details of the requirements; and to implement the record of the historical mappings. Relating to requirements prioritization, the suggestion is to offer a larger number of prioritization. For example, to adopt a scale from 0 to 100. Relating to the requirements mapping, it was suggested to implement the consistency of the definition of relationships between requirements, mainly in requirements with dependency, sub-requirements, and complementary requirements. It also was suggested to implement facilities to identify transitive dependency between functionalities.

Relating to requirements consultation, it was suggested to include a link to change the consulted requirement; to construct a dependency tree relating to requirements sources; to query deleted requirements; when consulting a specific requirement, to provide a link to visualize requirements data, presenting the changes history.

Finally, additional suggestions were proposed, such as: have a progress status monitoring for the requirements; place the types of requirements as IEEE standards; when the environment is used in security projects, insert attributes of ISO 15408; insert the link to the document source of the requirement; in the requirement register screen, to create a link to the requirement source; to delete the mapping link in the menu, keeping only the mapping change.

5 CONCLUSIONS

This paper presented DoMaR, an approach with an environment to assist in the requirements registration, organization and monitoring. The approach aimed to prevent some problems related to requirements management and documentation that occur during the execution of the RE process. A survey of the problems of requirements management and documentation was performed, so that the environment functionalities can cover and prevent such detected problems. An application was conducted using the environment, composed by eleven experts in the RE area. After analyzing the results, the conclusion, a priori, is that the environment features / functions have the potential to prevent the problems reported in the survey. Some suggestions given by the specialists, such as, include a link to change the consulted requirement, when consulting a specific requirement, to provide a link to visualize requirements data, presenting the changes history and have a progress status monitoring for the requirements will be implemented in the environment. The others remain for future work.

REFERENCES

ALEXANDER, I.; STEVENS, R. **Writing better requirements**. London: Pearson, 2002.

BOEHM, W. B.; PAPACCIO, P. N. **Understanding and controlling software costs.** IEEE Transactions on Software Engineering, v. 11, 1988.

ELECTRICAL, I. O.; ENGINEERS, E. **Standard glossary of software engineering terminology.** Los Alamitos, 1990.

FIRESMITH, D. **Specifying good requirements.** Journal of Object Technology, v. 02, n. 4, 2003.

FIRESMITH, D. **Common requirements problems, their negative consequences, and industry best practices to help solve them. Journal** of Object Technology, v. 06, n. 1, 2007.

HULL, E.; JACKSON, K.; DICK, J. *Requirements engineering*. 2. ed. London: Springer, 2005.

LEFFINGWELL, D. **Calculating the return on investment from more effective requirements management**. Cutter IT Journal, v. 10, 1997.

LIKERT, R. **A technique for the measurement of attitudes**. Archives of Psychology, v. 140, 1932.

SOMMERVILLE, I. **Engenharia de software**. 8. ed. São Paulo: Pearson Education, 2007.

VERNER, J. et al. **Predicting good requirements for in-house development projects**. In: INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING, NY, 2006.

YOUNG, R. R. **The requirements engineering handbook**. London: Artech House, 2004.

WIEGERS, K. E. **Software requirements**. Washington, DC: Microsoft Press, 2003.