A Set of Artifacts and Models to Support Requirements Communication Based on Perspectives
Early-stage Doctoral Research

Ana Carolina Oran
Institute of Computing – Federal
University of Amazonas (UFAM)
Country Code 69025-000
Manaus – AM – Brazil

ana.oran@icomp.ufam.edu.br

ABSTRACT

The effective communication of the requirements influences the success of software development projects. Achieving effective communication of the requirements is difficult due to the involvement of several persons with different roles, skills, knowledge and responsibilities. Although many studies analyze the communication between clients and system analysts, they do not focus on the communication within the development team. In this research, we propose the creation of a set of artifacts and models to support the communication of requirements. We will base our proposal on the different perspectives of the team members according to their experience in the artifacts and models adopted in the development process within their organization. We will follow a methodology based on *Design Science Research* guidelines, which will guide us through the creation and evaluation of the artifacts and models to solve problems with the communication of requirements. Our goal is to improve the communication of requirements between the members of a development team, reducing the loss of requirements information during the execution of the software project.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – *Methodologies, Languages*. D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *Documentation.*

General Terms

Documentation, Experimentation, Human Factors, Standardization.

Keywords

Effective Communication – Model experience assessment – Requirements Engineering Process – Requirements Communication – Design Science Research.

1. INTRODUCTION

Requirements engineering is a process that aims at identifying the stakeholders and their needs, and documenting them in a way that enables the analysis, communication and subsequent implementation and maintenance of a system [20]. Coughlan and Macredie [5] highlight that the requirements engineering is a difficult process because of the high complexity, vastness and volatility of the requirements. In this sense, requirements engineering is an important factor for the success of software development projects [14]. The cost and quality of the software depends critically on appropriate requirements engineering [3].

During the requirements engineering process execution, clients must be able to communicate their needs to the requirements analysts while the latter need to be able to communicate these requirements clearly and effectively to all of the other members of the development team. An effective communication of requirements is essential in all software projects [21] since there is a need for understanding the information throughout the software development process cycle.

Requirements engineering is crucial for the success of the project. Problems in requirements engineering can lead to serious consequences during the software development lifecycle [12]. Therefore, many measures have been proposed to ensure the quality of the software requirements specification [16]. In that context, Mund *et al.* [16] argue that the requirements specification document is the basis for the communication between the *stakeholders* in the requirements engineering. However, Liskin [11] states that this document can not be the only means of requirements communication, since each member from the development team has different information needs to perform his/her activities within the project.

Several researches are concerned with the requirements communication between client and analyst context [4][5][18]. However, these researches do not check the requirements communication among the development team of a software project. Therefore, to support improved requirements communication between the members of a software development team, this research proposes the creation of a method to assess the experience of development team members (Team Model eXperience - TMX) concerning the artifacts and models they use for communicating requirements during the software development project. This research also intends the creation of a set of artifacts and models to support the requirements communication based on the perspectives of development team members, such as developers, architects, testers, designers and others.

We defined the following research questions: RQ1: What are the main problems found by software development team during the communication of requirements? RQ2: Do the different role within the software development team have different needs regarding the requirements communication? RQ3: What are the most adequate artifacts and models for software development team during the requirements communication? RQ4. Which are the most frequently used artifacts and models by software development team during the requirements communication? RQ5. How does each member of the software development team think and feel about the artifacts and models they use during the requirements communication?

This paper is organized as follows: Section 2 presents the work related to the requirements engineering process, requirements communication in the development process, related work on User eXperience (UX) and Developer Experience (DX), and the contribution of the research to the state of the art. Section 3 presents the research proposal, the methodology to be employed for achieving the research goals and the threats to validity for this research. In Section 4, we present the current progress of the research to achieve the results. Finally, Section 5 describes the expected contributions.

2. RELATED WORK

2.1 Requirements Engineering Process

The main goal of the requirements engineering process is to find requirements with quality that can be implemented in software development. The requirements engineering is a process with the purpose of identifying the needs of users and documenting them in a way that it enables their analysis, communication and subsequent implementation [17].

The main artifact within the process requirements engineering is the requirements specifications document, which is a formal statement of requirements for the stakeholders. Such document provides the basis for the requirements analysis, requirements validation by the stakeholders, definition of what designers have to build, and verification if what was requested was properly developed at the time of delivery [17].

Although the requirements engineering is a well-established discipline, with a range of proposed approaches, techniques and tools [6], still there are many difficulties involved. Reasons behind these difficulties include that requirements engineering is complex by nature, because this is an interdisciplinary area in software development process that is driven by uncertainty and is therefore highly volatile and complex [13]. Additionally, the requirements process involving several views, roles, responsibility and goals for its implementation [19]. According to Méndez Fernández *et al*. [15], critical requirements engineering problems are related to communication issues and to incomplete/hidden or underspecified requirements.

2.2 Requirements Communication

Al-Rawas and Easterbrook [1] describe a field study regarding the communication of problems between the *stakeholders* in the process of requirements engineering within a software development project. Their results show that organizational and social issues have great influence on the effectiveness of the communication activities. The authors point out that the end users have difficulty in understanding and validating the requirements because of the notations used. Furthermore, software practitioners reported that the use of ratings that are understandable by clients (and often the use of natural language) create long documents and bring ambiguity. To deal with these problems, the development team makes use of face-to-face communication with the systems analyst.

Tu *et al.* [22] state that the use of more transparent requirements documents (i.e. with greater visibility of information to stakeholders) can contribute to a more effective communication. The concept of "Transparency" suggested by the authors is used to convey the sense of visible and open information to those who participate and need it. The documents transparency cited by the authors is based on three attributes: accessibility, understandability and relevance. Accessibility is the degree to which stakeholders can obtain information. Understandability is defined as the degree to which that information can be comprehended with prior knowledge. Relevance is the degree to which the information obtained by the stakeholders answers their questions.

The authors conducted an experiment with students and software practitioners who used two types of requirements documents with different levels of transparency. The result of the experiment conducted by Tu *et al.* [22] showed that there is evidence that the transparency based on accessibility attributes, comprehensibility and relevance can help evaluate the effectiveness of the documents as a form of communication. Therefore, it supports the theory that a higher transparency of requirements documents leads to a more effective communication.

Bjarnason *et al.* [2] report the results of an exploratory case study to a deeper understanding of the causes and effects of requirements communication failures in a software project. To do this, they applied semi-structured interviews with nine practitioners in a large software company. As a result of the study, they found four main factors that affect the communication of requirements: scale, temporal aspects, common perspectives and decision structures. The results also showed that the requirements communication failures lead to failure to meet the client's expectations, quality issues and wasted work effort.

Liskin [11] conducted a study in order to verify how requirements artifacts such as specifications, diagrams or *user stories*, can support activities related to requirements communication and requirements engineering. To do this, they also applied interviews with twenty-one practitioners in order to check how the artifacts were used by team members with different functions and different requirements informational needs. The interviews showed that a variety of artifact types is often required to successfully perform a software project. At the same time, using multiple artifacts can cause problems like manual translation effort and inconsistencies.

2.3 UX and DX

Many researchers have been concerned about the user experience. Consequently the studies in the User eXperience (UX) field are increasing. Fagerholm and Munch [8] define UX as a concept which captures how people feel about products, systems and services. According to the ISO 9241-210 [9] UX is the perception and answers of a person resulting from the use or anticipation use of a product, system or service. UX seeks to understand what the user speaks or thinks about a product, system or service. For instance, the user may report what was easy to use, what was confusing or whether it exceeded his/her expectations [23].

A similar concept to UX is the Developer Experience (DX) which is influenced by the "user experience" (UX) concept. DX aggregates relevant aspects, is intuitive, and helps organizations analyze, design or improve products or services. DX aims to help practitioners to better understand, analyze, design and improve the project environment with respect to the perceptions and feelings of the developers [8]. DX consists of checking how the developers think and feel about their activities within their work environments. According to Fagerholm and Munch [8], improving the developer experience can cause a positive impact on the software development project.

With the results achieved with the preliminary research of related work, we noticed that current work are more concerned with engineering requirements at the client and analyst level, not verifying what is the best way to communicate the requirements to the software development team. Considering such research gap, this paper proposes the creation of a set of artifacts and models to support the requirements communication, based on the perspectives of the members of the development team (e.g. architects, testers, designers, others) in order to increase the effectiveness of the communication throughout the project life cycle (i.e. analysis, design, development, testing and maintenance).

To achieve the improvement of the artifacts and models used by members of the development team for reporting requirements, we will elaborate an evaluation of the artifacts and models experience by members of the development team (Team Model eXperience - TMX), influenced by UX concepts and DX. By doing so, we intend to gather the strengths and weaknesses of these artifacts and models, identifying the needs for each role to perform the activities within the development process.

3. RESEARCH GOAL

Our proposed solution will deal with two aspects: (a) the evaluation of the experience of the members of the software development team within a project regarding the creation, validation and use of artifacts and models for the communication of requirements during the development cycle; and (b) the improvement of the effectiveness of communication of system requirements between the members of the software development team within the project, by proposing artifacts and perspective-based models.

The first aspect is based on the concepts of UX and DX, which are concerned with the evaluation of the end-user and developers experience in isolation and with regards for the product, systems and services. In this sense, our proposal seeks to create an assessment of artifacts and models experience by the development team (Team Model eXperience - TMX) regarding the creation, validation and use of the information generated from the requirements specification.

The second aspect is derived from the results presented in the works of [1], [2], [11] and [22], which state that both end users and software professionals have difficulties regarding the validation and understanding of the information generated from the requirements specification. In this sense, our proposal seeks to create artifacts and models that support the specification of requirements based on the perspectives of the consumers of this information.

As a final result, we intend to evaluate the experience of the team members regarding the use and maintenance of this information on software projects as well as develop a set of artifacts and models to support the requirements communication based on perspectives. These artifacts and models aim to support the creation, evaluation and distribution of documents generated in the requirements specification.

3.1 Methodology

For the development of this research, we will apply a methodology based on the guidelines of the *Design Science Research* method [24]. Design science is the design and investigation of artefacts for useful purposes [24] as illustrated in Figure 1.

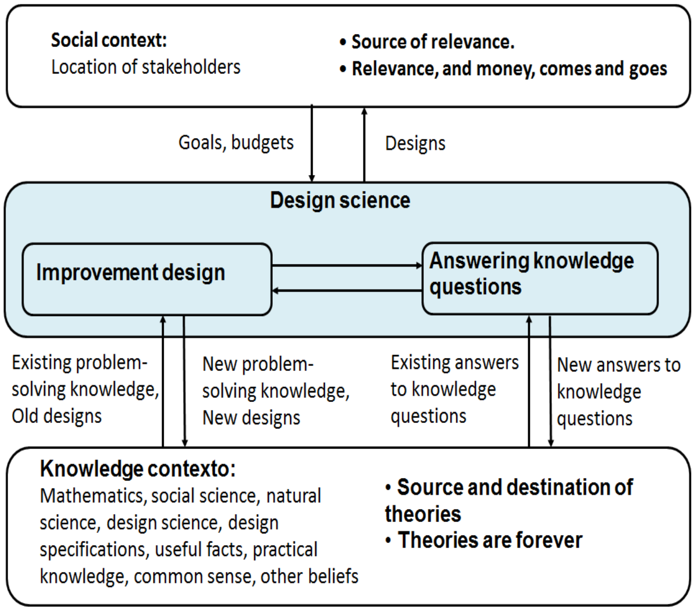
Social context:
Location of stakeholders
• Source of relevance.
• Relevance, and money, comes and goes
Goals, budgets
Designs
Design science
Improvement design
Answering knowledge questions
Existing problem-solving knowledge, Old designs
New problem-solving knowledge, New designs
Existing answers to knowledge questions
New answers to knowledge questions
Knowledge contexto:
Mathematics, social science, natural science, design science, design specifications, useful facts, practical knowledge, common sense, other beliefs
• Source and destination of theories
• Theories are forever

Figure 1. Framework for design science [24].

The social context (at the top of Figure 1) contains the stakeholders who may affect the project or may be affected by it. In our research context, the stakeholders are the members of the development team.

There are two types of research problems that can use the Design Science method: 1) designing an artifact to improve a problem in a given context and 2) answering questions about artifacts in a given context, as illustrated in Figure 1 (middle part). The knowledge context (the bottom of Figure 1) provides the theories existing about science and engineering, frameworks, instruments, constructs, models, methods and artefacts used in the develop/build phase.

The methodology used in our research is illustrated in Figure 2. It shows how we design and evaluate artifacts in our given context. In this research, we will explore the two types of problems. First, we will answer questions about problems that occur with the use of artifacts and models used in communicating requirements within development teams. To do this, we will use the TMX assessment and then we will create artifacts and models in order to improve this problem during software development.

Based on the Design Science approach, we aim to improve requirements communication in software development teams through artifacts and models, making them customized to the teams' needs in order to have a more effective requirements communication.

3.1.1 Stage 1

To identify gaps in the state of the art to propose new research opportunities, we adopted the first stage of the investigation of the problem within the methodology presented in Figure 2.

We performed an exploratory research regarding communication problems of requirements in software development projects. We selected relevant work in the area. As a result of this research, we obtain a set of initial works to be used as basis in the application of the Snowballing method. To select these papers, we used Google Scholar as it is a good alternative to avoid bias [25]. We applied the Forward Snowballing technique to search for papers citing our initial list of studies as a reference and Backward Snowballing to search for new papers in our list of references [25].

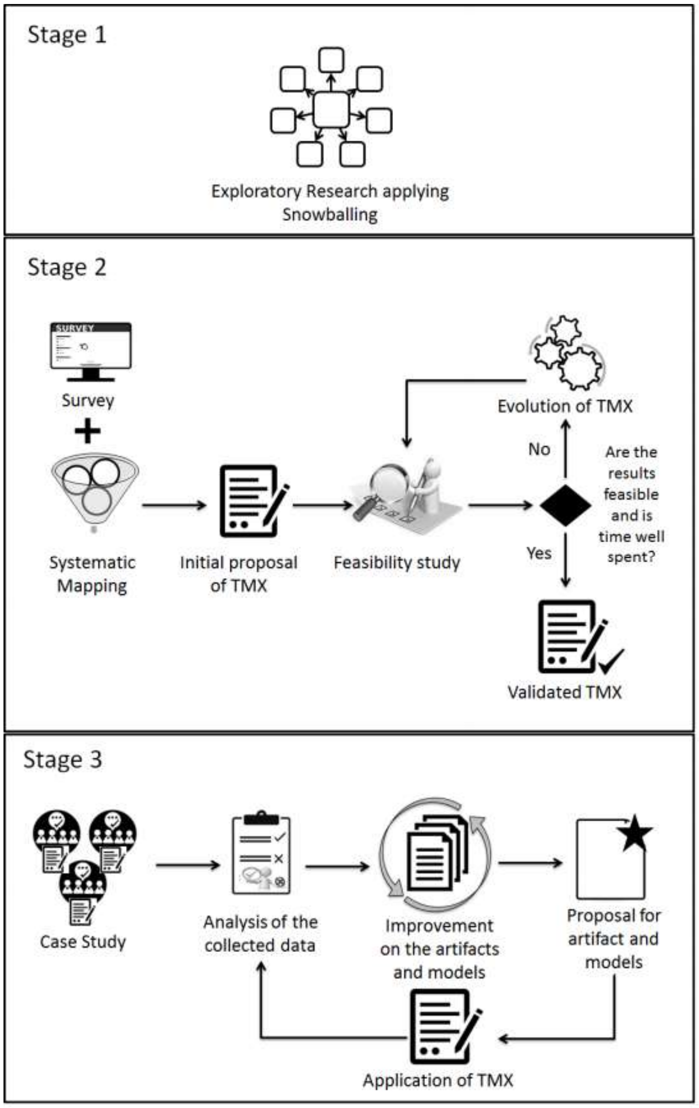
Stage 1
Exploratory Research applying Snowballing
Stage 2
SURVEY
Survey
Systematic Mapping
Initial proposal of TMX
Feasibility study
Evolution of TMX
No
Are the results feasible and is time well spent?
Yes
Validated TMX
Stage 3
Case Study
Analysis of the collected data
Improvement on the artifacts and models
Proposal for artifact and models
Application of TMX

Figure 2. Research Methodology.

3.1.2 Stage 2

To support the development and validation of TMX, the second stage of the methodology presented in Figure 2 will be adopted.

In the first activity, we will conduct a systematic mapping of the literature (SML) [10] to identify and analyze issues on the experience of the members of the development team regarding the creation, validation and use of information generated from the requirements specification. This SML will also identify features of other artifacts and models used for the requirements communication within the development teams. Simultaneously, we will execute a survey with professionals working in software development industries. This research will aim to find which artifacts and models are used by development teams for communication requirements, and which problems related to reporting requirements are faced by development teams. This information will be used as basis for development of new artifacts and models. After that, we will create the initial proposal of the experience of the members of the development team evaluation (Team Model eXperience - TMX) regarding the artifacts and models used by them for communicating requirements during the software development project. Then, we will conduct a feasibility study with software development teams using artifacts and models for communicating requirements, in order to assess the feasibility of using TMX and improve it. Depending on the outcome of the feasibility study, if needed, we will evolve TMX. Finally, we will recheck its feasibility until we find evidence that it is feasible. To verify TMX's feasibility, we will create minimum requirements for the validation of artifacts and models.

3.1.3 Stage 3

To develop of a set of artifacts and models to support the requirements communication based on the perspectives of the members of the development team, the third step of the methodology presented in Figure 2 will be adopted. Initially, we will perform case studies [26] in different software development companies, where we will apply the TMX and carry out interviews. These interviews will seek information regarding the experiences of the development team members on the artifacts and models used within the company to support the communication of requirements during the software project life cycle. After that, we will analyze the collected data to verify the negative and positive aspects of the used artifacts and models per role within the development team. With the analyzed data, we will suggest improvements in the artifacts and models applied in business. After that, we will be proposed new artifacts and models based on the data analysis. This proposal will be presented to the participating companies and will be applied again to TMX to evaluate the development team's experience in the new proposed artifacts and models.

3.2 Threats to Validity

In every empirical study, there are threats that can affect the quality of the results [7]. The main threats to validity considered during the execution of the studies from this research are:

(1) The differences between the subjects with regard to experience: The subjects may have different experience with the artifacts and models used in the requirements communication inside the organizations. To mitigate this threat, we will distribute the subjects by roles.

(2) Representativity of the subjects: the members of the selected development teams participating in the studies will be part large software development companies in the state of Amazonas (Brazil).

(3) Requirements communication: this research will be developed for the communication of requirements that were previously elicited. It will not focus on the techniques and results of the requirements elicitation phase with the client.

(4) The creation of the search string for the systematic mapping study: to identify synonyms for the creation of the search string, we will conduct an exploratory research applying the snowballing technique to collect the most relevant work in the area. By doing so, we will identify the keywords that can compose the search string. In addition, we will conduct tests to verify the adequacy of the search string on various digital libraries, checking if it returns the baseline papers identified on exploratory research.

(5) Inaccuracy in the extracted data and errors in the inclusion and exclusion of the papers: papers extraction and classification will be conducted by two researchers. The Kappa statistical test will be employed to assess the level of agreement between the two researchers. After that, we will carry out meetings to resolve discrepancies.

4. Current Research Status

The research started in March 2015, the first stage of the research problem was carried out using the Forward Snowballing and Backward Snowballing techniques. Currently we will start a systematic mapping within step 2 from our methodology to create TMX. This will assist the identification and analysis of the issues on the experience of the development team members regarding the creation, validation and use of information generated from the requirements specification. By doing so, we will identify a theoretical basis for the development of new artifacts and models.

5. EXPECTED CONTRIBUTIONS

This paper presents an initial proposal for this research that aims to develop a set of artifacts and models to support the requirements communication based on the different perspectives of development team members. To do so, we will conduct an evaluation of development team members regarding their experience on creating, evaluating, using and maintaining the resulting materials from requirements specification in order to improve the artifacts and models used for the requirements communication. At the end of this research project, we expect to achieve the following results:

- To identify the main documents, problems and improvement opportunities regarding the communication of requirements for software development;
- To identify the informational needs of requirements for each role inside the software development team;
- To create an method to evaluate the experience development team members regarding the creation, validation and use of artifacts and models used for the requirements communication during the development cycle;
- To define a set of artifacts and models to support the requirements communication based on different perspectives of the members of the development team;
- To decrease the time to clarify the incomplete/hidden or underspecified requirements.
- To decrease the time with rework and product fails due to requirement communication problems during the development cycle;

We expect that software development organizations employ TMX to check the experience of their development teams in the artifacts and models used for the communication of requirements within a project. Also, by doing so, we expect that these companies get an overview of what to improve in their artifacts and models and, therefore, increase the chances of success of their software projects.

6. ACKNOWLEDGMENTS

We acknowledge the financial support granted by FAPEAM (Foundation for Research Support of the Amazonas State) through processes numbers 062.00600/2014; 062.00578/2014, CAPES for the financial support to the first author of this paper, FAPERJ and UNIRIO for the financial support to the second author of this paper.

7. REFERENCES

[1] Al-Rawas A, Easterbrook S. 1996. Communication problems in requirements engineering: A field study, In: *Proceedings of Conference on Professional on Awareness in Software Engineering*, London, 1–2 February, 47–60.

[2] Bjarnason, E., Wnuk, K., & Regnell, B. 2011. Requirements are slipping through the gaps -A case study on causes & effects of communication gaps in large-scale software development. In

Requirements Engineering Conference (RE), 2011 19th IEEE International. 37-46. IEEE. DOI = http://dx.doi.org/10.1109/RE.2011.6051639.

[3] Broy, M. 2006. Requirements engineering as a key to holistic software quality. In *Computer and Information Sciences–ISCIS*. 24-34. Springer Berlin Heidelberg DOI = http://dx.doi.org/10.1007/11902140_3.

[4] Chakraborty, S., Sarker, S., & Sarker, S. 2010. An Exploration into the Process of Requirements Elicitation: A Grounded Approach. *Journal of the Association for Information Systems*, v. 11, n.4, 212-249.

[5] Coughlan, Jane; Macredie, Robert D. 2002. Effective communication in requirements elicitation: a comparison of methodologies, *Requirements Engineering Journal*, v. 7, n. 2, 47-60. DOI = http://dx.doi.org/10.1007/s007660200004.

[6] Daneva, M., Damian, D., Marchetto, A., & Pastor, O. 2014. Empirical research methodologies and studies in Requirements Engineering: How far did we come?. *Journal of systems and software*, v. 95, 1-9. DOI = http://dx.doi.org/10.1016/j.jss.2014.06.035.

[7] Easterbrook S., Singer J., Storey M.A., & Damian D. 2008. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer London, 285-311. DOI = http://dx.doi.org/10.1007/978-1-84800-044-5_11.

[8] Fagerholm, F., & Münch, J. 2012. Developer experience: Concept and definition, In *Proceedings of the International Conference on Software and System Process*, 73-77.

[9] ISO 9241-210:2010. International Standardization Organization (ISO). Ergonomics of human system interaction -Part 210: Human-centered design for interactive systems. Switzerland.

[10] Kitchenham, B., Dyba, T., Jorgensen, M. 2010. The value of mapping studies – A participant-observer case study. In *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*. British Computer Society. 25-33. DOI = http://dl.acm.org/citation.cfm?id=2664372.

[11] Liskin O. 2015.How artifacts support and impede requirements communication. In *Requirements Engineering: Foundation for Software Quality*. Springer, 2015, pp. 132–147. DOI = http://dx.doi.org/ 10.1007/978-3-319-30282-9_21.

[12] Mafra, P. ; Kalinowski, M. ; Méndez Fernández, D., Felderer, M. , Wagner, S. 2016. Towards Guidelines for Preventing Critical Requirements Engineering Problems. In: *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Limassol, Chipre. pp. 1-6.

[13] Méndez Fernández, D., Wieringa, R. 2013. Improving requirements engineering by artefact orientation. In *Product-Focused Software Process Improvement*, p. 108-122. Springer Berlin Heidelberg. DOI = http://dx.doi.org/10.1007/978-3-642-39259-7_11.

[14] Méndez Fernández, D., Wagner, S. 2015. A Case Study on Artefact-Based RE Improvement in Practice. 114-130. Springer International Publishing. DOI = http://dx.doi.org/10.1007/978-3-319-26844-6_9.

[15] Méndez Fernández, D.; Wagner, S.; Kalinowski, M.; Felderer, M. ; Mafra, P. ; Vetro, A. ; Conte, T. ; Christiansson, M. ; Prikladnicki, R. 2016. Naming the Pain in Requirements Engineering - Contemporary Problems, Causes, and Effects in Practice. In: *Empirical Software Engineering Journal*, Springer.

[16] Mund, J., Méndez Fernández, D., Femmer, H., & Eckhardt, J. 2015. Does Quality of Requirements Specifications matter? Combined Results of Two Empirical Studies. In *Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on* .1-10. IEEE. DOI = http://dx.doi.org/10.1109/ESEM.2015.7321195.

[17] Nuseibeh, B., & Easterbrook, S. 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*. 35-46. ACM. DOI = http://dx.doi.org/10.1145/336512.336523.

[18] Pacheco, C., & Garcia, I. 2012. A systematic literature review of stakeholder identification methods in requirements elicitation. Journal of Systems and Software, v. 85, n. 9, 2171-2181. DOI = http://dx.doi.org/10.1016/j.jss.2012.04.075.

[19] Pandey, D., Suman, U., & Ramani, A. K. 2010. An effective requirement engineering process model for software development and requirements management. In *Advances in Recent Technologies in Communication and Computing (ARTCom)*, 2010 International Conference. 287-291. IEEE. DOI = http://dx.doi.org/10.1109/ARTCom.2010.24.

[20] Pires, P. F., Delicato, F. C., Cóbe, R., Batista, T., Davis, J. G., & Song, J. H. 2011. Integrating ontologies, model driven, and CNL in a multi-viewed approach for requirements engineering, *Requirements Engineering*, v. 16, n. 2, 133-160. DOI = http://dx.doi.org/10.1007%2Fs00766-011-0116-1.

[21] Stapel, K., Knauss, E., & Schneider, K. 2009. Using flow to improve communication of requirements in globally distributed software projects. In *Requirements: Communication, Understanding and Softskills, 2009 Collaboration and Intercultural Issues*. 5-14. IEEE. DOI = http://dx.doi.org/10.1109/CIRCUS.2009.6.

[22] Tu, Yu-Cheng; Tempero, Ewan; Thomborson, Clark. 2015. An experiment on the impact of transparency on the effectiveness of requirements documents. *Empirical Software Engineering*, v. 21, n. 3, 1035-1066. DOI = http://dx.doi.org/10.1007/s10664-015-9374-8.

[23] Tullis, T.; Albert, W. 2008. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

[24] Wieringa, R. 2009. Design Science as Nested Problem Solving. In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, Philadelphia, 8. ACM. DOI = http://dl.acm.org/citation.cfm?doid=1555619.1555630.

[25] Wöhlin, C. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*. n. 38. DOI = http://dx.doi.org/10.1145/2601248.2601268.

[26] Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A. 2000. Experimentation in Software Engineering: An Introduction. Boston, MA: Kluwer Academic Publishers.