

Requirements Engineering with a Perspective of Software Evolution

Anticipating requirements based on organizational change

Marília Guterres Ferreira¹, Julio Cesar Sampaio do Prado Leite¹

¹Pontifícia Universidade Católica do Rio de Janeiro, PUC - Rio,
Brasil

{mferreira, julio}@inf.puc-rio.br

Abstract. Software Evolution is a fact in industrial life. More than that, maintenance is one of the most expensive phases on software life cycle. Additionally, social and organizational aspects are increasingly gaining greater importance for information systems development. In this context, Organizational Semiotics has been considered a promising tool, providing the framework MEASUR for requirements gathering. In this work, it has been studied how to apply Organizational Semiotics in Requirements Engineering with the purpose of Software Evolution.

Keywords: Requirements Engineering, Software Evolution, Organizational Semiotics, Anticipating Requirements, Organizational Change.

1 Introduction

Software evolution is a fact. Most of the changes in software-based systems are caused by changes in organizations from the need to adapt to a more and more competitive and dynamic environment. Those changes in software are far higher than the initial development costs. If when we develop new software systems, we try to anticipate the ways in which they might change, then the software can be modified easily to implement the new requirements. As most changes in software come from organizational change, to be able to predict how the organization will evolve, we should first define what its current state is and then determine for which state it will go. The objective of this research is to understand the way organizational changes impacts system requirements in order to anticipate requirements that come from software evolution.

The authors are members of the Requirements Engineering Group (ER Group), a group of academic research in Requirements Engineering, and also of the Software Engineering Laboratory of PUC-Rio (LES PUC-Rio), a laboratory of academic research applied to the development of software solutions for industry. Most projects developed in LES deals with software evolution and in some cases the software has to be developed from scratch because the maintenance on the running system is very

difficult and consequently expensive and time-consuming. This research comes from the realization that some of the problems could be avoided if some requirements could be elicited in advance. Observations in the cases have found that the requirements that evolved could be classified in *technological* (related to the evolution of the technology used: programming language, hardware, peripheral systems) or in *organizational* (related to the evolution of the business domain). The system barely can be previously prepared to technological evolution, but when it comes to organizational evolution, it may be not only prepared but also cause and anticipate the changes. The software evolution projects showed a need of requirements engineering more geared to social, political, cultural and ethical aspects.

In this context Organizational Semiotics seems to be a promising theory. Organizational Semiotics is a discipline that studies the use of signs and their effects on social practices. On this subject, there is Stamper's School [1] which proposes a set of methods, MEASUR research program, to the design of information systems based on the social-technical paradigm, considering social, political, cultural and ethical issues involved in understanding the problem in the process of requirements engineering. In Brazil, following this school, Baranauskas and colleagues [2], [3] propose a semioticbased method for stakeholders identification and requirements elicitation. Study cases showed that activities from Organizational Semiotics carried out deal with information not captured by other techniques, involving cultural, behavioral, ethical and political aspects [2], [3]. These points make Semiotics relevant to Requirements Engineering and to the problem addressed in this work.

The purpose of this work is to develop a strategy for Requirements Engineering with a perspective of Software Evolution based on the Organizational Changes. Other recent researches are carried on anticipation of requirements and are been studied and the grounding of this study. Pimentel and colleagues [4], [5] presented foresight techniques to predict requirements for autonomic systems. Rolland and colleagues [6] propose to model changes as a set of gaps between the requirements specification of the current and of the future system. This work aims is to unify the aforementioned researches with a perspective of Organizational Change in order to identify future requirements to make Software Evolution process less arduous. This is an incipient work and its references are its most important related works.

2 Objectives of the Research

The general objective of this research is defining a strategy to anticipate requirements change and consequently make the software evolution less traumatic, less expensive and in less time.

The specific objectives are:

1. Identify organizational concepts influenced by software;
2. Identify organizational concepts that influence software;
3. Define trends of requirements change in a software evolution process;
4. Merge all aforementioned and define a conceptual framework of organizational characteristics most likely to change to be considered in software evolution.

For this, we propose the method showed in Figure 1. This is a preliminary method since it is still on study.

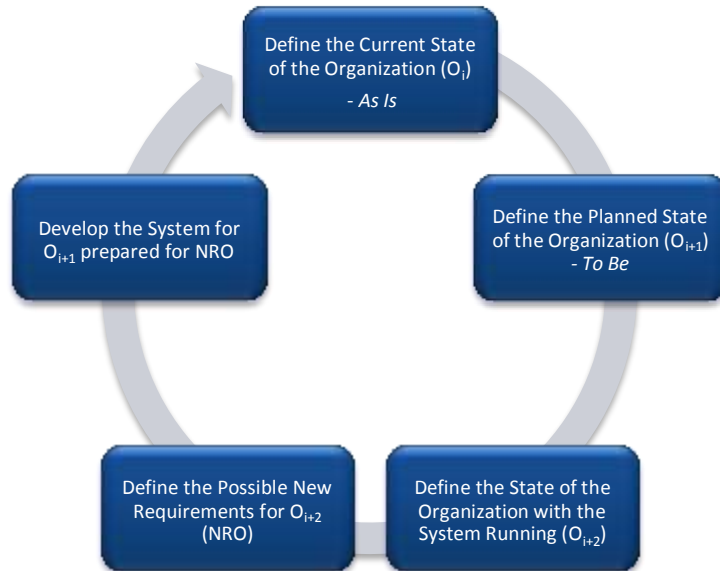


Figure 1. Preliminary Method for Requirements Engineering with a Perspective of Software Evolution

This Preliminary Method is explained as follows:

1. **Define the Current State of the Organization (O_i) –As Is:** In this phase, the focus of the requirements engineers is on model the current characteristics of the organization. Here, the problems to be addressed by the software are defined. This step is supported by the method PAM (*Problem Articulation Method*) of Organizational Semiotics.
2. **Define the Planned State of the Organization (O_{i+1}) – To Be:** In this phase, the organization is modeled as planned to be. It is time to elicit the goals, the functions and the constraints that the software must to address. This step is supported by the methods SAM (*Semantic Analysis Method*) and NAM (*Norm Analysis Method*) of Organizational Semiotics.
3. **Define the State of the Organization with the System Running (O_{i+2}):** Once the system is running, it implies in some changes in the organization’s culture. This phase concentrates on model the changes and differences between O_i and O_{i+1} . Theories from Organizational Change Management help this step. This step is also supported by the method PAM of Organizational Semiotics.
4. **Define the Possible New Requirements for O_{i+2} (NRO):** With the model O_{i+2} in mind, define what would be the new requirements for this organization. This step is also supported by theories from Organizational Change Management and by the methods SAM and NAM of Organizational Semiotics.

5. **Develop the System for O_{i+1} prepared for NRO:** the system to be developed will address the current requirements, elicited in steps 1 and 2, but it will also be prepared for the requirements anticipated by steps 3 and 4 for the evolution process to be less traumatic, less costly and in a shorter time.

Figure 2 shows the disciplines underlying the suggested method and the main points of each step.

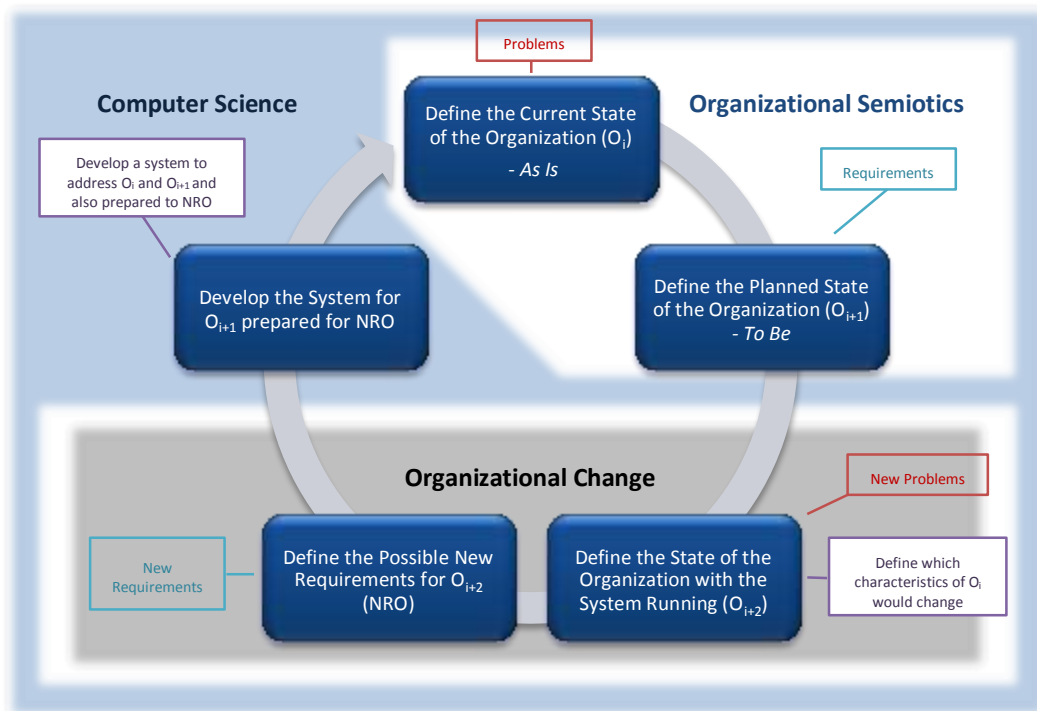


Figure 2. Disciplines that orient the Preliminary Method for Requirements Engineering with a Perspective of Software Evolution

3 Scientific contributions

The main contribution of this work is the definition of a strategy to requirements engineering with a perspective of software evolution. This strategy aims to anticipate requirements change based on organizational change. For this, first we define the current state of the organization in two steps, first the problems and then the semantic and norms, based in methods from Organizational Semiotics, PAM, SAM and NAM respectively [1], [2]. And then, based on the changes that the system might cause in the organization, the steps 3 and 4 define the future state of the organization, i.e., the new problems and the new requirements, again supported by Organizational Semiotics. The software-based system to be develop should address the current needs and be prepared for the ones anticipated, thus the software evolution will be less difficult.

4 Conclusions

Software evolution is a recurrent fact in industry and usually under intense time pressure. The anticipation of requirements might support this process. The strategy presented in this study is incremental, once the current model is defined, the next steps determine what is needed to be added, excluded or modified. It is an epistemic tool that aims to help in the design of the problems and requirements of the organization, either in the present and in the future. The expected results are to make the software evolution process less difficult and in a shorter time. It is an ongoing work and further research is required to improve and validate the proposals.

5 Ongoing and future work

Currently, we are investigating real industry cases of software evolution. The preliminary method will be applied in each organization in a *retrospective way*, i. e. looking into the past and into the present, in order to verify if we could have predicted the new requirements with the method. The steps of the analysis to be made will be explained as follows:

1. **Model the past state of the organization:** through conversations with the requirements engineers of the project and research on artifacts and documents, we will elicit the business policies and rules and the problems in the organization that were addressed with the system. It is modeled based on the past, based on PAM. This will be the O_1 .
2. **Model the desired stated of the organization:** in this point, we will model what was desired in that moment for the organization to be, what were the requirements. Also modeled based on the past, based on SAM and NAM. This will be the O_2 .
3. **Define the current state of the organization:** then, we will model the organization in the present, its new processes, business policies and rules and, mainly, its new problems. We will define how the organization is with the system running. It is modeled based on the present and again in PAM. This will be the O_3 .
4. **Define the current requirements for O_3 :** now, we will elicit the new requirements for the organization with the system already running. It is modeled based on the present and ever again in SAM and NAM. This will be the NRO.
5. **Analyze the system evolution:** In this point, we will compare the “old” and the current (NRO) requirements and analyze if the NRO requirements could be previously identified if the preliminary method was applied on that moment.

Through the study of the cases, we intend to verify the applicability of the preliminary method with current software evolution projects, once the prediction of requirements may demand long time to be validated. This study will help to better define and improve the method.

Moreover, through the analysis of the nature of change in requirements, we also intend to identify if there is a pattern on requirements evolution, that is, if we could define a framework for requirements evolution. Once the framework is defined, we have to verify if we can point the hotspots on it that should be taking into account in software evolution processes.

As a future work also to be tackled in this thesis research, we pretend to verify if we can also model a conceptual framework of what are the points in an organization that can evolve and at the same time are related to software, points that can either influence the software and be influenced by it. This framework can point out which are the conceptual hotspots that usually evolve and should be considered in any elicitation process. Then, merge the two aforementioned, i. e., make a framework that points at the same time, the organizational characteristics that influences the software and are more likely to change in the process of software evolution. For this we intend to base our researches on theories arising from Organizational Change Management.

References

1. Liu, K.: *Semiotics in Information Systems Engineering*. Cambridge, England: Cambridge University Press (2000).
2. Baranauskas, M. C., Schimiguel, J., Simoni, C. A., & Medeiros, C. B.: Guiding the Process of Requirements Elicitation with a Semiotic-based Approach – A Case Study. *Proceedings of the 11th International Conference on Human-Computer*, 3, 100-110 (2005).
3. Simoni, C. C.: *A Prática de Desenvolvimento de Software e a Abordagem da Semiótica Organizacional*. (Dissertação de Mestrado) Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), Campinas (2003).
4. Pimentel, J., Santos, E., Castro, J., & Franch, X.: Anticipating Requirements Changes Using Futurology in Requirements Elicitation. *International Journal of Information System Modeling and Design (IJISMD)*, 3, pp. 89-111 (2012, April).
5. Pimentel, J., Castro, J., Perrelli, H., Santos, E., & Franch, X.: Towards anticipating requirements changes through studies of the future. *Fifth International Conference on Research Challenges in Information Science*, 1–11. doi:10.1109/RCIS.2011.6006858 (2011)
6. Rolland, C., Salinesi, C., & Etien, A.: Eliciting gaps in requirements change. *Requirements Engineering*, 9(1), 1–15. doi:10.1007/s00766-003-0168-y (2004).