Towards a Requirements Engineering Process for Innovative Software: Learning with the *ConecteIdeias* Case

Renata Guizzardi[1,2], Marcos Accioly[1], Letícia P. Fonseca[1], Roberta L. Gomes[1], Miriam M. Pinto[1]

[1]Laboratory of Technologies to Support Collaboration Network
[2]Ontology & Conceptual Modeling Research Group
Federal University of Espírito Santo
[3]Centro de Referência em Formação e em Educação a Distância
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo
Vitória, Brazil
rguizzardi@inf.ufes.br, marcosaccioly@gmail.com, leticia.fonseca@ufes.br, rgomes@inf.ufes.br, miriam.pinto@ufes.br

Abstract. For many years, Requirements Engineering has been regarded as a crucial phase of software development, because it is responsible to collect from the stakeholders and model the purposes of the system to be. Consequently, lots of research results have been accumulated over the years concerning approaches to elicit and model requirements. However, eliciting and modeling requirements for innovative software, for which no determined stakeholder is defined, are still open research issues. This is due to the very nature of innovation, which asks for new ways of thinking and generating software products. This paper aims at presenting the RE process which led to the development of a web-based social network named *ConecteIdeias*. This social network is aimed at supporting people on creating new ideas and solutions to change social reality in the place where they live. Besides briefly presenting the social network and thoroughly describing the process, this paper also presents some lessons learned and good practices gathered throughout the process execution. The long term goal of this research is to propose a general RE process for the development of innovative software.

1 Introduction

The velocity and complexity of software development processes especially aiming at innovation is ever growing these days [1]. Thus, being successful in innovating depends, more and more, on the ability of interacting and exchange knowledge. In other words, in the past few years, we have been watching a change in paradigm in software development, coming from more to less structured processes, especially concerning the determination of the purposes of the software to be and, thus, in the Requirements Engineering (RE) context.

In general, RE has been appointed as a crucial phase in software development, being considered by renowned researchers like Boehm [2] as well as practitioners as the reason for either successful software (when properly done) or for huge failures (otherwise). However, if we consider the aforementioned paradigm shift, we must expect the need for drastic changes in the RE practices.

Finkelstein [3] claims that today, the model for software adoption is analogous to the one of ***TV channel selection***. Users download and delete new software apps and adhere and remain in this or that social network system, depending on their immediate use and perception. Thus, software developers have more and more difficulties in attracting and especially in maintaining customers.

When one considers developing software to support the needs of specific organizations or stakeholders, the RE process is somehow facilitated by the fact that the requirements engineers have someone to consult, question, negotiate, i.e. interact when developing the system to be. In other words, RE is less complex when the development occurs on demand. In other cases, however, there is no a priori contractor. In such cases, the developers themselves imagine that there is an unmet need for a software system. Thus, from the point of view of the developer, the client is not yet known and he is not sure if the need is real. This is what we call ***innovative software***. In the case of innovative software, market and product are both unknown at the beginning of the development process. In these contexts, novel functionalities must be developed and not simply copied from a software or context to another. Otherwise, there is no innovation at all.

Having set before the reader the main characteristics and difficulties in developing innovative software, a question that begs for an answer is: ***what is an appropriate RE process to develop innovative software?*** This research question is the long-term goal of this present research. In this paper, we describe the RE process applied in a recent project aiming at developing an innovative software, namely the *ConnecteIdeias* web-based social network[1]. Besides describing the process lifecycle and activities, we also highlight some best practices and lessons learned during the process execution. It is important to highlight that, when we say best practices and lessons learned, we are merely using the Knowledge Management jargon for positive and negative impressions respectively, as opposed to empirically validated claims.

ConecteIdeias's top goal is to provide social and technological innovation to empower communities to improve different social, environmental, and economic issues affecting them, by the constructive debate around ideas (i.e. solutions to these issues). To support system development, we chose an economically challenged community in the city of Vitória/ES (Brazil) as *ConecteIdeias's* target user community. Selecting a user has been crucial to enable us to understand the requirements and to test the system's prototypes, by developing several experimentation throughout the RE cycle.

Aiming at clearly describing the contributions of our work, the remaining of this paper is organized as follows: Section 2 brings background information, along with the analysis of some related work; Section 3 presents some initial information about

[1] http://www.conecteideias.com

the case study; Section 4 describes the *ConecteIdeias* social network; Section 5 presents the RE process used in the case; and Section 6 concludes the paper.

2 Background Information and Related Works

2.1 Innovative Projects

Innovation is all about being able to create new value through new products, services, production processes or social arrangements [1]. Such new value can be economic, social or environmental. It can be made using markets or not, but innovation always requires multiple actors. In other words, innovation is necessarily a collective effort. It also involves doing things differently from what is already being done and to do so, it requires a project structure.

Projects are unique, one-time initiatives, assembled to be temporary and to achieve a specified goal under the constraints of time, budget and other resources. However, innovative projects have specificities. Particularly, they differ in terms of novelty degree, availability of technologies required, complexity and pace [4].

Novelty refers to how new the product or service is to the market, the customers, the potential users, i.e. to the society in general. Novelty is particularly relevant because it defines the process of defining the product's or service's requirements. According to Wheelwright and Clark [5] in terms of novelty, new products and services are classified as derivatives, platforms or breakthroughs. Derivatives are extensions of existing products or services while platform projects correspond to developing and producing new generations of existing products and services. Breakthrough projects aim at introducing a totally new product or service, based on new concepts or new technologies. Thus, novelty corresponds to a beneficiary's perspective.

Design methodologies to elicit the possible future customers and beneficiaries requirements are helpful [6] as well as the understanding of business models that can make the new product or service viable and sustainable [7]. As the level of definitions is low at the beginning of the project and it lowers as the project runs, requirements must remain flexible until first market introduction is made and until customer's feedback is available.

Fast prototyping is very important. Moreover, working close to potential or desired customers or beneficiaries is critical and one of the tasks to get success is to educate them about the potential of the new product or service and to articulate hidden needs [4].

2.2 Related Works: Requirements Engineering for Innovative Software

Research on processes and methods to support RE for innovative software seems to be in its early stages in the academic community. Most of the works we have found do not refer to innovation. They do however explore Design techniques especially the ones supporting creativity to enable requirements elicitation and analysis.

Recently, Lemos et al. [8] conducted a systematic mapping study on creativity in RE. As a result of this study, four groups have been considered as the main drivers of this research area: a) Maiden/Robertson et. al; b) Berry et. al; c) Nguyen et. al; and d) Schimd et. al. We believe that different aspects of each of them may be useful to our research but none of them share the same focus.

From Maiden/Robertson et. al and Berry et. al, we can profit from experimenting with the proposed and applied techniques for requirements elicitation. In fact, some of the techniques we applied in the early phases of our RE process are very similar to the ones applied in Maiden/Robertson et. al. The works of Berry et. al on the other hand focus on the proposal of a method called EPMCreate, based on the model of the pragmatics of communication. EPMCreate still relies on the existence of specific stakeholders for the system-to-be and would require a big adaptation for application in cases similar to the one we describe here.

The works of Nguyen et. al can contribute with a more theoretical view of creativity for RE. In this respect, more recent works such as [9] may add interesting points to this discussion as it presents a study to help us understand the meaning of creativity used in a specific RE context to help us decide about the required support for such context.

Moreover, Horkoff and Maiden [10] have also made some interesting progress on the use of *i** to support creativity in RE. Their idea is to overcome problems of the free-form representation of creative workshops Maiden previously applied, by taking advantage of goal analysis to support the decision making in RE.

Although many interesting works have been found, the analysis of such works has shown that the existing RE practices for generating innovation are far from consensual and many points are open for discussion, including the proposal of a RE process such as the one we are currently investigating.

3 General Information about the *ConecteIdeias* Case

ConecteIdeias was funded by a state funding S,T&I agency as an innovation project, aimed at putting together academia, private sector and society in the development of a new social platform. The main innovative feature of *ConecteIdeias* is to provide interest groups with an online tool able to help them transforming ideas into reality.

In principle, all actors were committed to develop an innovative system. One of the first actions we took was registering the software-to-be, giving ownership of the software to all actors in Table 1, which also present these actors' responsibilities.

We attribute many of the successes of the project to the multidisciplinary nature of the project team. Regarding the academic actors, this created many knowledge exchange opportunities and motivated these actors to engage in the project's activities.

Throughout the project, carrying out responsibilities was not as smooth as one would like. For example, many times, the Requirements Engineer and Junior Researcher on Industrial Design had to take over the implementation; the Researcher on Innovation had to manage the project, etc. One important lesson learned is for next time, to use some consensus making method to help the team agree on some core concepts regarding the project from start, for instance the concept of innovation itself.

We discuss more about this on Section 5, where we describe the RE process and highlight the best practices and the lessons learned throughout the process execution.

Table 1. *ConecteIdeias* developing team actors along with their specific responsibility

Actor	**Specific Responsibility**
Researcher on Innovation	Inform the team about innovation principles and serving as the "team glue", managing the actors.
Researcher on Project Management	Manage the development project.
Senior Researcher on Industrial Design	Using and conducting activities with the appropriate Design techniques throughout the system development. For example, User-centered Design and Creativity techniques.
Researcher on Collaborative Software and Requirements Engineering (Computer Science Background)	Supervising the RE activities, support on conceiving and modeling Collaborative Software
Researcher on Collaborative Software and Mobile Applications (Computer Science Background)	Supervising the development of the android version of *ConecteIdeias*, support on conceiving Collaborative Software
Requirements Engineer (Computer Science Background)	Requirements analysis and modeling
Junior Researcher on Industrial Design	Conducting activities with the appropriate Design techniques throughout the system development. And developing the system's interface design.
Company Senior member	Contribute in the RE activities and supervise the implementation of the system-to-be. The company was also in charge of hosting the social network servers.
Four Company Developers	Implement the System-to-be.

4 The *ConecteIdeias* Social Network

The *ConecteIdeias* social network allows anyone to create an idea while the other users can interact with him/her, helping to improve or even to implement it. In other words, the system supports people sharing a common goal (regarding a specific city issue) to expose, discuss and possibly accomplish their ideas. For that, *ConecteIdeias* needs to support people not only in proposing a solution in the virtual world (i.e. the web), but also in pursuing such solution in the real world.

ConecteIdeias has been developed as a web platform where any registered user can connect to. In order to register themselves, users must enter their name, address and email. Once connected, the user is taken to the *ConecteIdeias* main page which presents the **Ideas Wall**, as shown in Figure 1. This wall displays the ideas created by the whole community, ordered by creation date, number of concerned people, or number of comments.

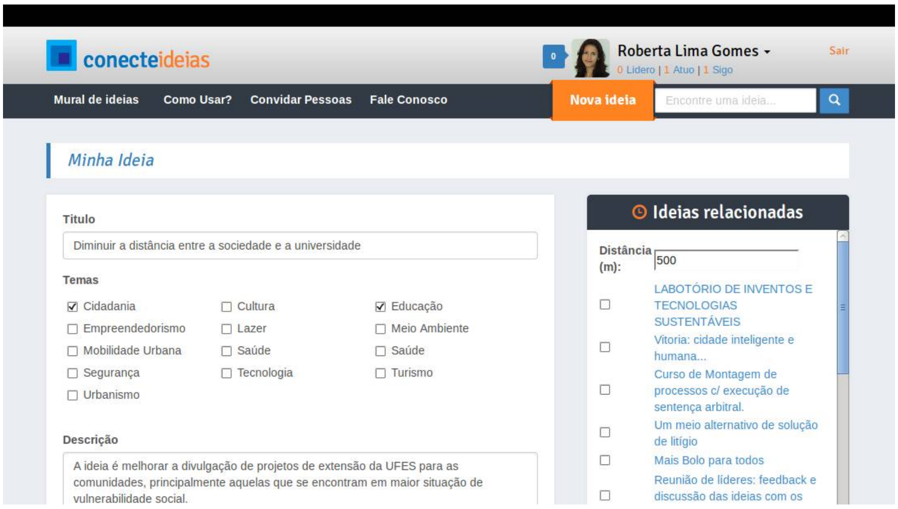


Figure 1. *ConecteIdeias* main page after login

The ***idea*** is then the project’s central element. One idea is essentially composed of a theme, a descriptive text, images and geographic location. Besides the panel seen at Figure 1, there is an alternative screen, where people are able to interact with ideas directly on a map, also viewing the **active** and **accomplished ideas** according to their geographic location.

Any registered user can create an ***ideia*** and then become its **leader**, as illustrated in Fig. 1. A user can also collaborate with already created ideas. In order to do so, first he/she needs to visualize an idea, either in the **Ideas Wall** or by using any navigation facility provided in the platform. For example, while viewing an ***idea***, it is possible to view its **related ideas**. Users can also search for a specific theme or even a specific user, and then view the ***ideas*** related to this user.

Users can **act** on different ***ideas*** leaded by other users, provided that they receive an invitation by the **idea’s leader** or **active user**. When **acting**, users are able to send comments or images. Otherwise, users can also **follow** an idea. In this case, the user may only passively observe other users’ interaction concerning that idea. While **acting** or **following** an idea, users can also invite other users to participate on that idea.

Commenting and placing images are not the only way to act on an idea. Real world actions (that might lead the idea being accomplished) may also be registered. These actions are called **ongoing action**, and they are presented as highlighted comments. Accordingly, users are able to have an overall view of the **idea's evolution**. Once the idea has been accomplished, the leader can finally set it as **accomplished**.

Usually, people that live on socially excluded communities do not have access to personal computers, often accessing the Internet through a mobile phone. Thus, besides the web-based application, a mobile application has also been developed (on Android Platform) in order to facilitate system's accessibility. Moreover, the software development company that implemented *ConecteIdeias* has also developed a Microsoft-based version of the system, so as to support their in-house clients.

5 The RE Process Used to Develop *ConecteIdeias*

In this Section, we present the RE process we applied to develop the *ConecteIdeias* Social Network. Figure 2 depicts the RE process lifecycle.

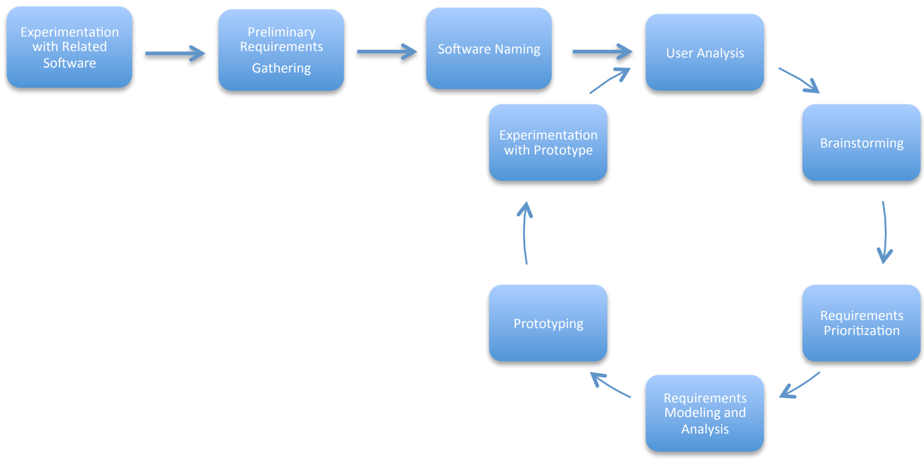


Fig. 2. *ConecteIdeias* RE Process Lifecycle

As shown in Fig.2, the process was carried out by executing three sequential activities, followed by performing a few cycles, each one composed of six iterative activities. In what follows, we describe all these activities.

A. Experimentation with related software. We chose three social networks that are also aimed at serving as a collaborative platform around problems noted by citizens about their city, i.e. systems that shared the goals with the software under development[2]. Then, we made some experimentation with the targeted stakeholders to gather some impressions about their relation with the kind of technology required by the system-to-be. This activity was conducted by the Industrial Design researchers in our team. For this activity, the general public of the community was involved: 13 people (regarding age: six in the 20-30 range, six in the 13-19 range, and one 40 year-old participant).

To contrast with the public involved in the first workshop, we also organized an experimentation workshop with the members of an Aiesec chapter. Aiesec is an organization aimed at developing young leaders, generally targeting University undergraduate students or recently graduated students. This workshop gathered 4 people, with age ranging from 18-30 years old. In this workshop, besides the Industrial Design researchers, a senior member of the development company that implemented *ConecteIdeias* was involved.

As expected, the results of both workshops were different. The Aiesec group proved to be more familiar with social networks, finding it easy to use the chosen

---

[2] MGov2 (http://mgov2.org), My fun city (http://myfuncity.org) e Porto Alegre (http://www.portoalegre.cc).

systems. During the experimentation, they were able to suggest enhancements to the software in use.

The community group, on the other hand, presented some difficulties while trying to use the systems. They were afraid of doing something wrong (which showed us the need of developing a system that provides constant feedback to the user); they could not read the words in English, like tags, for instance; it took them a long time to navigate through the social network and, many times, they did not understand the goals of the actions they were supposed to undertake in a specific webpage within the system in use. As a result of this experimentation, we also found out that they are very used to Facebook and, consequently, they are used to upload photos and videos, besides linking and sharing information.

The distinct results obtained in the two workshops led us to consider, from the start, that we should develop the same functionalities in different ways of interaction, so that we would support different types of users. But despite the different results, there were also some commonalities. Both groups mentioned the suggestion to facilitate the creation of ideas, because in the tested software, this functionality was hard to find and the tutorials were too long and boring. They also both highlighted the need to have filters for finding ideas from the same topic and the same place. This feature was also missing in the tested software. And finally, they have also pointed out the importance of using different colors to emphasize and classify information.

Best Practice

It is very interesting to test with software that shares goals with the system-to-be. This way, many Do's and Don'ts are learned from the experimentation with similar features.

It is a good idea to experiment with groups of users with distinct profiles, because they will provide different feedback based on their level of experience with the kind of software being developed. And moreover, you will find out some general requirements from the feedback in common.

B. Preliminary Requirements Gathering (PRG). We invited some community leaders for a workshop with the aim of gathering the first requirements for the system-to-be. For this activity to be effective, it was important that the interaction with the system became something concrete. Thus, the activity was elaborated with basis on paper, using cards that represented the inputs and outputs of the system. Please refer to Figure 3, which shows scenes of this activity. Some examples of requirements that emerged in this workshop are: a) attaching images and videos to the posted ideas; and b) printing banners and other publicity material generated as a result of the debate regarding a posted idea.

Best Practice

It is essential to simulate the interaction with the system-to-be by using concrete material. Humans are used to concrete things and are liable to provide you with more useful feedback when facing concrete interaction.

(i) (ii)

Figure 3. (i) paper cards and (ii) scene from the PRG Workshop

C. Software Naming. Originally, the system-to-be was called *Mosaic*. But after activity B, one of the community leaders pointed out that such a name did not make the system's proposal explicit. Every once in a while, we were invited to participate in a community leaders' forum meeting to present our proposal. Thus, we took in the advice regarding the name and, in one of these meetings, we asked the leaders to provide the system-to-be with a new name. For that, a brainstorming was conducted, resulting in the proposal of several interesting names. Then, by vote, the leaders chose *ConnectIdeas* as the name of the Social Network under development.

Best Practice

Relying on the stakeholders to make decisions during the software RE process has a good impact in engaging people around the goals of the innovation project.

D. User Analysis. The contact with the targeted stakeholders helped us understand several difficulties people had with the use of websites and apps, in writing text and etc. Thus, before each cycle of requirements prioritizing, modeling and prototyping, we conducted an analysis, which helped us raise some points of attention that we reminded people throughout the whole cycle, so as to come up with solutions to cope with such problems.

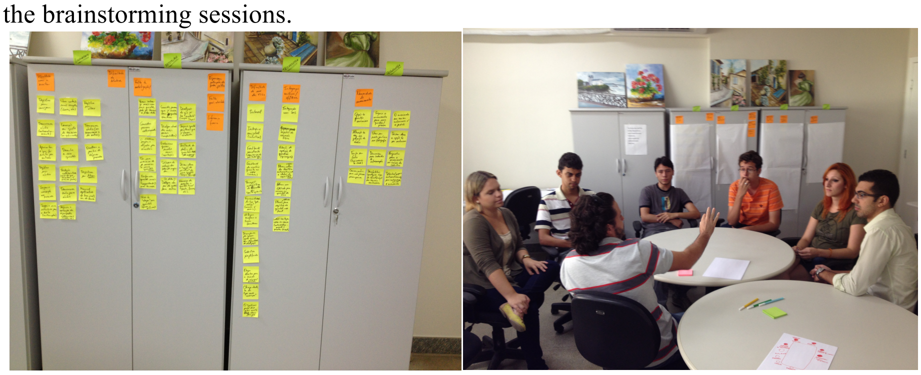
For the first user analysis activity, we used the information of the activities A-C, which turned out to be very valuable in terms of requirements gathering. In general, we concluded that we had to create different ways to ease the users interaction with the system. For instance, a) creating a tutorial composed of a lot of images and not some many pieces of text, helping people to intuitively learn how to use the system; b) to provide templates with some predefined text to support writing; c) using distinct colors to highlight and classify information. Besides these points of attention, a general reminder resulting form the outcomes of activity A was to allow different types of interaction with the system, so as not to bore more experienced users.

Throughout the cycles, this activity was carried out a number of times, always guiding us to look at the user as the target when conceiving new system's requirements. An interesting functionality type that was targeted as a result of this activity was a way for the system to gather different types of statistical information regarding the use of the distinct features of the system, so as to provide us a way to analyze the user, even after system release.

Best Practice

Satisfying the software user remains as the main focus of the RE activity. However, as the user is not always present and available for requirement elicitation, performing analysis of user characteristics and statistical data is paramount to feed the requirement engineering cycle with useful knowledge.

E. Brainstorming. We organized some brainstorming sessions to elicit requirements from the LabTAR members, mostly composed of students and researchers in the fields of Computer Science, Industrial Design and Production Engineering. A few actors from the company's development team were also present in a couple of meetings (junior members). Two of the Industrial Design researchers were responsible for conducting the brainstorming sessions. In these opportunities, we proposed different themes (e.g. How to connect the discussed ideas? How to make the system user friendly? How to make the system more intelligent? How to turn the discussed ideas at *ConecteIdeias* something tangible in the real world?). For each theme, people could freely suggest new features whenever a new idea came to their mind. Figure 6 depicts the brainstorming sessions.



(i) (ii)

Figure 6. (i) Post-its registering requirements; (ii) group involved in one of the brainstorming sessions.

In this activity, there was no censorship, i.e. all requirement proposals were welcomed and registered in post-its. After each brainstorming session, we gathered the post-its and classified the requirements, creating different hierarchies for each theme. This classification meant to facilitate the requirements engineers to understand and set requirements in priority order, thus supporting the next activity in the RE cycle.

Best Practices

Brainstorm with the development team can be helpful to support people in thinking out of the box, arriving at innovative requirements.

The more multidisciplinary the brainstorming team, the better, because a professional from a field inspires the others, creating a rich knowledge exchange.

While brainstorming, it is important not to censor any incoming ideas.

F. Requirements Prioritization. We took the resulting document from activity E to a session where we gathered the requirements analysts, the senior members of LabTAR and the senior members of the software development company, whose main responsibility was to implement *ConnectIdeas*.

The objective of this session was to set the requirements in priority order, also giving a chance to the development company's collaborators to add new requirements when needed. No particular requirements prioritization technique has been applied in this activity, although we acknowledge that such methods can be beneficial. The choice for not using one was rather a pragmatic one. The project's team was very heterogeneous and the time was not feasible to accommodate training of such techniques.

In practice, very few requirements were added. In fact, the company members acted more to restrict or censor the list of requirements, listing those of high priority that should be implemented, and often making clear that some of the requirements with less priority would never come into play.

In general, we noted that such censorship was not only due to prioritization or to make the development process more agile. Many times, restrictions happened as a result of a different view of the project and were often not interested in generating innovation, but rather using the opportunity to develop something they could sell to their clients using the established paradigm. As consequence, the different visions the industrial partners had when compared to the members of academia and user community demotivated and generated several disagreements between the development team. The practical impacts of such disagreements in *ConecteIdeias* adoption have been discussed in [11].

Lessons Learned

The team needs to reach a consensus regarding what Innovation is before the project starts. Otherwise, people will commit to different goals and this may hamper the development of innovative software.

No matter if the majority of the project members are engaged and oriented towards Innovation. One uncommitted partner is enough to undermine the project's goals towards Innovation.

Communication between academic and industrial partners is difficult because of the different work contexts and language. Thus, communication should be managed and cared for.

G. Requirements Analysis and Modeling. Having the requirements' list in priority order, the requirements analyst proceeded to analyze and model requirements. For each cycle, the analyst together with a RE researcher decided upon the scope of the model, with basis on the previous talks to the company's team (activity F). Then, the requirements analyst built an ER model. And at each cycle, the ER model was refined, incorporating concepts, relations and attributes.

ER was chosen because the requirements analyst had previous experience with this language while not having any with goal modeling languages or other RE frameworks. However, we do believe that more sophisticated techniques, such as goal modeling could be beneficial to this activity, since it provides more appropriate techniques to analyze and document the decision making process in RE. For instance, *i** [12] allows such analysis and documentation through decomposition and alternative analysis, criteria prioritization through contribution analysis, among others.

In this activity, there was usually lots of debate among the academic members and especially with the requirements engineer regarding whether or not to ignore the restrictions imposed by the industrial partner (see activity F). There is a high motivation to ignore these restrictions, since the current activity (G) was performed in the University lab. However, this has proven not to be such a good idea, especially since the next activity (H) is to be carried out by the industrial partner itself. Thus, more conflicts and miscommunications arose whenever we decided to ignore the agreements made on activity F.

Best Practices

At each cycle, it is very important to define the scope of the RE model, so as to incrementally develop the system.

Lessons Learned

It is paramount to deal with members' anxiety and different expectations regarding the system to-be, because in such process there is no system owner. Thus, the modeled requirements have to result from consensus in activity G. Otherwise, more conflicts and miscommunication are bound to happen among the team members.

H. Prototyping. As prescribed by the researchers on innovation (refer to Section 2.1), *ConecteIdeias* was developed incrementally, by providing a more complete prototype at each cycle. Thus, it is difficult to separate the RE process from the development process altogether.

This activity was carried out by the company's developing team, although in some cycles, some members of LabTAR also got involved in coding. Wulpi and Pinto [11] present more detail about which parts of the system was implemented by which actor.

The collaboration among the different programmers was not very smooth. Besides the different views of the project (discussed in F and G), the fact that the programmers were geographically disperse and often working in different hours was also challenging. To support that, regular on site or skype meetings were carried out to assign responsibilities and update the information regarding the system modules. Moreover, some code management software were used, such as github[3] and the MS TeamFoundation[4] built for MS Visual Studio[5] (specifically used for *ConecteIdeias's* MS Windows version).

---

[3] https://github.com

[4] https://www.visualstudio.com/en-us/products/tfs-overview-vs.aspx

[5] https://www.visualstudio.com/

Best Practices

Code management software is an essential support to assist in collaboration, versioning and code reviewing, especially when the programmers are geographically disperse.

Lessons Learned

For all activities but especially to this one, it is very important to make good predictions regarding time/effort. Otherwise, this can activity may be a difficult bottleneck in the process.

It is paramount that it becomes clear who is going to implement the software and to get the commitment of the assigned partner to replace programmers if the task is not being carried out on time and under specified quality criteria.

I. Experimentation with the Prototype. The main goal of this activity was to test the implemented functionalities of *ConecteIdeias* prototype at each cycle. This was important 1) to gather feedback regarding whether these functionalities were up to the standards or needed some improvement; and 2) to gather new requirements regarding new functionalities from the target user.This activity was carried out following the same dynamics of activity A, and again by the researchers in Industrial Design. For pragmatic reasons, we could not test *ConecteIdeias* with other user groups as before.

Given the variety of more or less formalized approaches to RE, it is not easy to consider a standard RE process. We are indeed able to state that such process is usually composed at least of activities such as *Elicitation*, *Analysis*, *Negotiation*, *Validation* and *Documentation* [2], iterated in distinct ways, depending on the method. In a sense, the process we propose targets all these activities. Many of the differences in dealing with such activities are highligthed in the posed best practices and lessons learned, but others may be only read in between the lines. Unfortunatelly, for lack of space, we are unable to make these differences more explicit. We emphasize, however, that the main differences in our process are motivated by the fact that there is no specific targeted stakeholder, as in most systems developed for organizations.

6 Conclusions

In this paper, we describe the RE process used in an innovation project carried out to develop *ConecteIdeias*, a web-based social network aimed at supporting users in debating and implementing ideas to solve city problems.

It is our claim that developing innovative software requires a specific RE process, having in mind the specificities of generating innovation products. Here, we highlight and analyze some lessons gathered throughout *ConecteIdeias* RE process execution.

From start, the project team was committed to generating a breakthrough innovative product as described in Section 2.1. However, several issues regarding the interaction among the project team members led to the current version of the system, which we consider a derivative product.

We do acknowledge that we do not attempt to close the discussion in this regard, but rather to start one. Much work remains to be done before we can claim we have a steady and concrete process proposal. First of all, more empirical evidence is needed so as to prove the quality of the proposed process. Moreover, we must acknowledge the limitations of our work with dealing with documentation ambiguity and other issues that are so far out of the scope of this work but as important with respect to the development of innovative software. Thus, future work includes generalizing the RE process and applying it in other case studies. Moreover, our research agenda also includes trying out different Design-oriented techniques and related to improve this process and related challenges.

Acknowledgement. This work is partially supported by FAPES (grant number 56380372/2012) CAPES/CNPq (grant number 402991/2012-5) and CNPq (grant numbers 461777/2014-2 and 485368/2013-7).

References

1. Martin, B. R. The evolution of science policy and innovation studies. Research Policy, v. 41, n. 7, (2012), 1219--1239.
2. Boehm, B.: Requirements Engineering at Age 20: Looking Back, Looking Ahead. ICRE (1996).
3. Finkelstein, A.: The Next 10 Years: the shape of software to come and what if means for software engineering. In: Keynote at ER@BR (2013).
4. Shenhar, A. J., & Dvir, D. Reinventing project management (1st ed.). Boston, Massachussets: Harvard Business School Press. (2007).
5. Wheelwright, S. C., & Clark, K. B. Revolutionizing product development. New York: Free Press. (1992).
6. Brown, B. T., & Wyatt, J. Design Thinking for Social Innovation. Stanford Social Innovation Review, (winter), (2010), 29--35.
7. Blank, S., & Dorf, B. The startup owner's manual (1st ed.). Pescadero, California: K&S Ranch, Inc. (2012).
8. Lemos, J., Alves, C., Duboc, L, and Rodrigues, G.: A Systematic Mapping Study on Creativity in Requirements Engineering. In: 27th Annual ACM Symposium on Applied Computing, ACM, (2012), 1083--1088.
9. Mahaux, M., Mavin, A, and Heymans, P.: Choose your Creativity: Why and How Creativity in Requirements Engineering Means Different Things to Different People. In Regnell, B., and Damian, D. (eds.) Requirements Engineering: Foundation for Software Quality, LNCS 7195 Springer, Berlin (2012), 101--116.
10. Horkoff, J., Maiden, N., and Lockerbie, J.: Creativity and Goal Modeling for Software Requirements Engineering. In: 2015 ACM SIGCHI Conference on Creativity and Cognition, (2015), 165--168.
11. Wulpi, L., and Pinto, M.: Alinhamento no Planejamento e Flexibilidade na Execução são críticos para o Sucesso de Projetos de Colaboração Universidade-Empresa. (In Portuguese). LabTAR Technical Report, available at: http://www.labtar.net.br/site/producoes/ (2015)
12. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.): Social modeling for requirements engineering. MIT Press, Cambridge (2011).