Requirements Engineering Using Mockups and Prototyping Tools: Developing a Healthcare Web-Application

Leonor Teixeira[1,3], Vasco Saavedra[1],
Carlos Ferreira[1,3], João Simões[4], and Beatriz Sousa Santos[2,3]

[1] Department of Economics, Management and Industrial Engineering,
University of Aveiro, Portugal
[2] Department of Electronics, Telecommunications and Informatics,
University of Aveiro, Portugal
[3] Institute of Electronics and Telematics Engineering of Aveiro (IEETA), Portugal
[4] Truphone, Lisbon, Portugal
{lteixeira,vsaavedra,carlosf,bss}@ua.pt, jpasimoes@gmail.com

Abstract. Healthcare web-application development teams involve non-computer experts working (clinicians) on the requirements specification that is later processed by software engineers/analysts (conceptual model) and coded by software programmers (software project). The management of this process, which involves different levels of abstraction and professionals with different backgrounds, is often complex. As such, mediators and facilitator's mechanisms for the requirements-gathering process and information transfer are needed. The main purpose of this work is to minimize the problems associated with this complex process, supporting the requirements engineering process of a healthcare web-application in a rapid prototyping model. The results proved that a rapid and functional prototyping model can improve the effectiveness of the requirement elicitation of any software development.

Keywords: healthcare web-application, software, requirement engineering, requirement elicitation, mockups, prototyping.

1 Introduction

The success of any software depends on how well it fulfills the needs of its users and of its environment. Software requirement comprises these needs and Requirements Engineering (RE) is the process by which the requirements are determined, being a fundamental part of the development process of any software. The first stage of the RE process, commonly known as requirement elicitation, is recognized as one of the most critical activities of the entire development lifecycle, since it is the stage where the main stakeholders are identified and involved in order to specify, analyze, and define the software goal and respective software requirements. It is a negotiation process during which intense capturing, combination and dissemination of knowledge

S. Yamamoto (Ed.): HIMI 2014, Part I, LNCS 8521, pp. 652–663, 2014.
© Springer International Publishing Switzerland 2014

occurs, and different stakeholders exchange information about the context, and the tasks that will be supported by the software under development [1]. However, the involvement of stakeholders is not always an easy process, and the degree of their involvement depends on several factors, such as: (i) newness of the project; (ii) degree of complexity of the system; (iii) techniques and methods used by the analyst to promote the elicitation and validation the requirements, and; (iv) the geographical dispersion of potential users, which can difficult the joint meetings usually required to identify and validate the requirements.

Particularly in the health domain, the low rate of technology acceptance and the percentage of software projects that have failed in this area continue to be a phenomenon that has placed challenges to researchers that work in this knowledge area. The literature indicates that non-acceptance of a particular technology is frequently associated with the non-involvement of potential users in the development process, and a great part of the failed projects are due to the lack of systematic considerations of human aspects throughout the design process [2, 3]. Regardless of application domain of the software/Information System (IS), the involvement of the users, although important at all stages of the project, requires a special attention in the early stages of the process and should be mediated with a set of techniques and methods appropriate to the context and type of problem.

Accordingly, there are different methods and techniques that are used by system analysts to manage the RE process, such as UML models, task analysis, and prototyping. Within the scope of prototyping, there are a variety of technological solutions that allow the creation of mockups, being Lumzy[1] one of them. With tools like Lumzy, it is very easy to create mockups, share and send them to the stakeholders involved in design process in real-time, putting the emphasis on collaboration and interactivity, and promoting the clarification and validation of the software requirements, when the support team is geographically distributed.

The aim of this work is to present our perspective on the requirements engineering process of a health information system, as well as the lessons learned from our experience with the development of a distributed web-application to support the Portuguese National Registry of Hemophilia and other Congenital Coagulopathies (NRH&CC), using Lumzy as the prototyping tool. To present this subject, this paper is structured in four sections. In section 2, an overview of the related work in the requirements engineering and prototyping is presented. Section 3 presents our case study concerning the RE in a process of a health information system development using a prototyping model. Finally, in section 4, the main conclusions of the paper are presented.

2 About Requirements Engineering and Prototyping

In the software/Information System (IS) development context, a requirement is a property that a system must exhibit in order to meet the system's motivation need. A

[1] `http://www.lumzy.com/`

software requirement is a property which must be exhibited by the software developed to solve a particular problem within one specific context. Requirements Engineering (RE) is a science that studies, analyzes and documents the requirements, and is presented in the literature as a knowledge area of Software Engineering that specifies, analyzes, and defines the product goal and functionalities of the final solution [4, 5]. RE can be described as series of stages including elicitation, analysis, specification, validation, and management of the software requirements. During the first stage of the RE process, which is referred to as the requirement elicitation phase, systems requirements are discovered, discussed and agreed by the stakeholders. This stage is defined by Pohl [6] as a core RE activity aiming to determine relevant requirements sources, to identify functional and non-functional requirements from these sources and to discover new requirements. RE activities are vital in ensuring successful projects and shortcomings in requirements elicitation can have a negative impact on the overall development process and consequently can lead to higher costs for the involved organizations [7]. Furthermore, the task of requirements elicitation is highly collaborative and involves many stakeholders, including the users who interact with system and usually represent the domain experts.

According to Sommerville [8], the process of requirements engineering is difficult for several reasons, emphasizing the fact that: (i) in many cases the users/clients[2] are not completely sure about their real needs, and often don't know what they want from the system, except in general terms; (ii) users/clients express requirements in their own language and with implicit knowledge of their work, and requirements engineers without experience in the user domain must understand these requirements; (iii) different stakeholders have different requirements, which they may express in different ways.

To help the process of requirements engineering and minimize the difficulties inherent to this process, the literature presents a set of tools and methods to assist the organization of the requirements, consistency checking, preparation of the specification, and formalization and validation of these requirements.

Usually, requirements are obtained and documented in a natural language (list of textual requirements) after they are modeled using a formal requirements representation (e.g. UML models, task analysis) or figures (e.g. mockups) in order to validate the requirements. While techniques like interviews, questionnaires, user observation, workshops, brainstorming, card sorting, and think aloud are used to obtain the software requirements, UML/Use-cases, task analysis methods and prototyping are examples of tools to model these requirements, providing mechanisms to assist a possible validation. However, few of these tools and methods provide any valuable help for obtaining, representing and validating the requirements by potential users who are geographically distributed and have no availability to attend to team meetings.

A number of studies have considered prototyping as an excellent vehicle for requirements discovery and validation [9, 10], since the prototype model can help elicit the requirement from the changing and complex environment, decomposing high-level requirements into details. Prototyping is a popular requirements elicitation

[2] Non-technical stakeholders involved in requirements identification.

technique because it enables users to develop a real sense about final systems that have not yet been implemented [10]. Through the use of prototypes, users can identify the true requirements that may otherwise be impossible to identify, by visualizing the software systems to be built, being especially useful when there is a great deal of uncertainty about the requirements [11].

In the RE context, prototypes can serve different purposes, and may be classified as low-fidelity prototypes and high-fidelity prototypes [12]. The low-fidelity prototypes are those that do not resemble the final product, being widely used in the exploratory phase or in the early stages of system development. Usually this type of prototypes is simple, easy to produce and therefore involves low costs of production. The high-fidelity prototypes are closer to the final product and usually use the same techniques and materials in its development. This type of prototype has a much higher associated cost in comparison with the former type. Regarding the implementation, it is possible to classify the techniques of prototyping in two methods: throwaway or evolutionary prototyping. The former is the development of a prototype which aims to increase the quality of the requirements document and is generally based on the most complex requirements. This type of prototype is discarded after fulfilling its purpose. Examples of this type of prototype are the paper prototype and the wizard-of-Oz prototype [13]. The wizard-of-Oz prototype is often used in situations with complex functionalities or situations needing to test new ideas, simulating system responses according to user actions. Evolutionary prototypes, on the other hand, are developed as a portion of the actual system and focused on the requirements that have already been well understood and new requirements are incrementally added, as the development proceeds in an iterative manner.

To meet the needs in the context of RE, and taking into account the cost/benefit relationship of this process, low-fidelity prototypes of the throwaway type are the most used to obtain early feedback in uncertainty environments, being discarded after finalizing its purpose. In order to assist the RE process through this type of prototype, there are a variety of tools (desktop or online applications), that allow to draw mockups and create wireframes, such as Balsamiq, Visio Professional, OmniGraffle, Prototype Composer, ConceptDrawPro, SmartDraw, Pencil Project, MockFlow, fluidIA, Pidoco, or Lumzy. These tools include libraries with graphical elements, which enable managing and publishing information elements, as well as incorporating new graphical components [13]. A prototype generated by such tools have the same advantages as a paper prototype, having a low cost, easiness of construction and use, and providing an interface that allows the user to perceive that is using a disposable prototype. Additionally it inherits some useful features of the wizard-of-Oz prototypes, and even of the evolutionary prototypes, to the extent that it's possible to include interactivity, with actions responding to user events. With this type of tools, clients can design their own interface as they would like to use it, simply dragging components into the canvas. Some of these tools also have the particularity to be accessible online, thus not requiring any additional software installations by the user.

With Lumzy, the tool used in our case study, is very simple to create mockups, share and send them in real-time to the stakeholders involved in the design process. Lumzy is a web-based wireframing tool for rapid user-interface prototyping, emphasizing collaboration and interactivity, as well as promoting the clarification and validation of the software requirements without the presence of stakeholders in the same geographical space. Lumzy represents a powerful simulation software that enables users to rapidly build functionalities without writing code, offering both technical and non-technical users to experience, test, collaborate and validate the simulated program, also providing reports with annotations.

This paper reports a case where a rapid prototyping was adopted to elicit and validate the system requirements at a healthcare level and Lumzy was the tool used to mediate and manage the RE process.

3 Practical Application: Requirements Engineering Process in a Health Information System (IS) Using Lumzy Prototyping

This work aims to present the requirements elicitation process of a health information system (Health-IS) development, as well as the lessons learned from this experience. The Health-IS involved is a distributed web-application to support a National Registry of Hemophilia and other Congenital Coagulopathies (NRH&CC) in Portugal, and prototyping was the main technique used in the requirements elicitation.

3.1 Motivation and Overview of the Project (Health-IS)

The present project is a joint initiative between the hemophilia healthcare professionals, represented by the Portuguese Association of Congenital Coagulopathies (PACC) and a group of researchers from the University of Aveiro (UA) responsible for analyzing, developing and implementing the technological solution.

The lack of a NRH&CC in Portugal, associated with the difficulty that clinicians of this area faced in order to manage this specific patient information, motivated a group of physicians to search for a technological solution that allowed to facilitate and optimize the information management process. Thus, the need arose to develop a project that led to the creation of the NRH&CC in Portugal, and the PACC members constituted themselves as the main clients/users of this project, using their experience as a basis for the requirements definition process. The University of Aveiro (UA) was designated to develop the project, having the responsibility to conceptualize, encode and implement the technological solution (named *hemo@record*). Given the complexity of the project, which involved a broad range of demographic, social and clinical data, and the geographic dispersion of the users/clients responsible for defining the requirements (associated with the limited availability), the process of Requirements Engineering emerged as the main challenge of project.

3.2 Requirements Engineering Process Using a Prototype

The process of developing an application with the described features is always complex, as in addition to the development team members having very diverse backgrounds, they are geographically dispersed. In order to perform the process of requirements engineering of this project, we chose to use a mediator and facilitator method for obtaining and managing the requirements, specifically the throwaway prototyping method, which consisted of a development lifecycle model by which a prototype is created for demonstration and requirements elicitation. The sequence of the steps followed using this technique is described below and depicted in Figure 1.

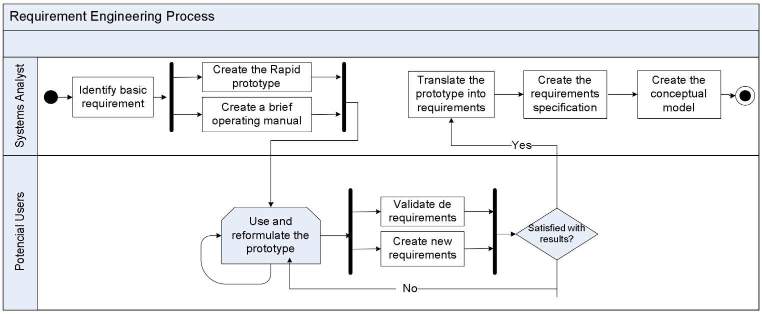
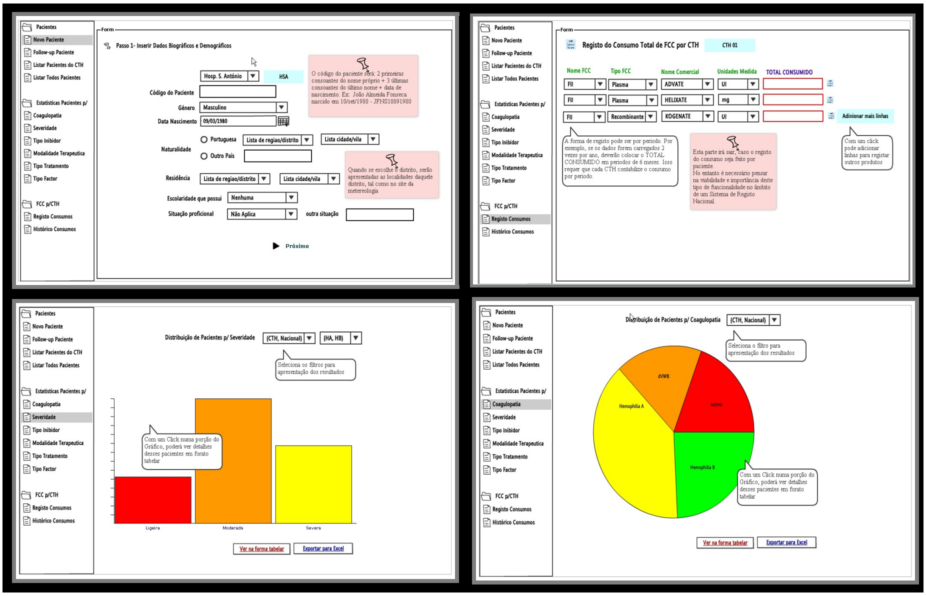

Requirement Engineering Process
Systems Analyst
Potencial Users
Identify basic requirement
Create the Rapid prototype
Create a brief operating manual
Translate the prototype into requirements
Create the requirements specification
Create the conceptual model
Yes
Use and reformulate the prototype
Validate de requirements
Create new requirements
Satisfied with results?
No

Fig. 1. Phases of the requirement engineering process

First Step (Start-up Meeting). The process started with a meeting attended by the members of PACC (users and clients of the project) and the members of the team responsible for the development, having been defined the objectives as well as the high-level requirements to include. This meeting produced a first draft of the Requirements Specification Document (RSD) in a text format, with some very high-level requirements.

Second Step (1st Version of Prototype Building). Given the high level requirements defined in phase 1, complemented with the results of a previous study of the authors [14] and based on the analysis of the national reporting systems already implemented in other countries, the systems analyst drafted the first version of the prototype (designated *prototype-hemo@record*), using the Lumzy prototyping tool. This version of the prototype included a set of requirements not yet validated by the stakeholders. The purpose for this inclusion was to stimulate the curiosity of the stakeholders, forcing them to validate and possibly to add missing requirements and/or delete irrelevant requirements. As can be seen in Figure 2, this version structured the information through a side menu, which according to the selection of the user allowed to access several interfaces for data entry or present the information displayed in different formats, descriptive text and graphics representations.

It should be noted that Lumzy, despite its easiness of use, can still cause some resistance from users without computational background if not properly contextualized and introduced in the environmental analysis. In order to eliminate this barrier, we prepared a small manual of instructions, which was sent to each participating stakeholder, with the access codes to use the *prototype-hemo@record* application. Eight invitations were sent to eight clinicians and future users of the technological solution.



Passo 1- Inserir Dados Biográficos e Demográficos

Hosp. S. António ▼ HSA

Código do Paciente

Género: Masculino ▼

Data Nascimento: 09/03/1980

Naturalidade: ○ Portuguesa Lista de regiao/distrito ▼ Lista cidade/vila ▼ ○ Outro País

Residência: Lista de regiao/distrito ▼ Lista cidade/vila ▼

Escolaridade que possui: Nenhuma ▼

Situação proficional: Não Aplica ▼ outra situação

O código do paciente será: 2 primeiras consoantes do nome próprio + 3 últimas consoantes do último nome + data de nascimento. Ex: João Almeida Fonseca nascido em 10/set/1980 - JFNS10091980

Quando se escolhe o distrito, serão apresentadas as localidades daquele distrito, tal como no site da metereologia

Registo do Consumo Total de FCC por CTH CTH 01

Nome FCC	Tipo FCC	Nome Comercial	Unidades Medida	TOTAL CONSUMIDO
FII ▼	Plasma ▼	ADVATE ▼	UI ▼	
FII ▼	Plasma ▼	HELIXATE ▼	mg ▼	
FII ▼	Recombinante ▼	KOGENATE ▼	UI ▼	

A forma de registo pode ser por periodo. Por exemplo, se os dados forem carregados 2 vezes por ano, deverão colocar o TOTAL CONSUMIDO em periodos de 6 meses. Isso requer que cada CTH contabilize o consumo por periodo.

Esta parte irá sair, caso o registo do consumo seja feito por paciente.
No entanto é necessário pensar na viabilidade e importância deste tipo de funcionalidade no âmbito de um Sistema de Registo Nacional.

Com um click pode adicionar linhas para registar outros produtos

Distribuição de Pacientes p/ Severidade {CTH, Nacional} ▼ {HA, HB} ▼

Seleciona os filtros para apresentação dos resultados

Com um Click numa porção do Gráfico, poderá ver detalhes desses pacientes em forato tabelar

Distribuição de Pacientes p/ Coagulopatia {CTH, Nacional} ▼

Seleciona o filtro para apresentação dos resultados

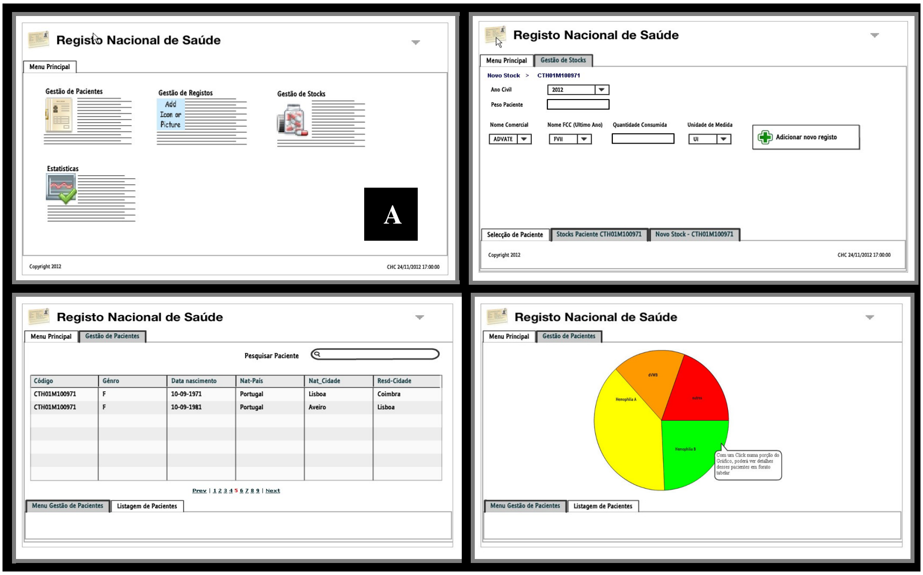
Com um Click numa porção do Gráfico, poderá ver detalhes desses pacientes em forato tabelar

Fig. 2. First version of the *prototype-hemo@record*: same examples of mockups (in Portuguese)

Third Step (1st Prototype Validation and Reformulation). After sending the invitations with the access credentials and instructions, the clients/users were encouraged to use and test the prototype, being able to change, remove or add new features. This phase took place over three months, involving an iterative and incremental process of evaluation and redesign of the prototype. As the prototype was modified, the analyst gathered the main features and the necessary data for defining the conceptual model and, at the same time, was able to understand the main difficulties of the users. During this phase, the analyst also had an important role in the maintenance of the prototype, ensuring the organization of the elements (icons) that were placed by the users on the interface, and, in some situations, converting the requirements inserted by the users in the annotation format, in actual functionalities of the prototype. Based on this prototype version, the functional requirements were identified. In terms of non-functional requirements, the analyst became aware of difficulties in using the prototype by some users, more specifically on the usage of the navigation system. The information did not appear to be easily found by users, causing them to replicate on certain pages, the information already presented on others. The identification of these difficulties prompted the need to rethink the organization of the information and the navigation mechanism, thus resulting in a new version of the *prototype-hemo@record*.

Fourth Step (2nd Version of Prototype Building). This version included all the requirements already identified in the previous version, and a completely restructured layout and navigation mechanism (Figure 3). At this stage, and taking into account all the requirements previously identified, the analyst, in collaboration with the developers of the system, proposed a new navigation mechanism, specifically oriented to processes. Each information process would be presented in the first interface (Figure 3 – Interface A), with the sub-processes being presented in the subsequent interfaces. Thus, all the information not relevant to the process involved would be hidden.

After this restructuring, a new version of the prototype was submitted to be evaluated by the same group of eight users, in order to validate the functional requirements in the new layout, while at the same time, testing some non-functional requirements. The access to the previous version was maintained, to allow to perform direct comparisons with the new version, thus allowing the users to indicate the version they found more intuitive and easier to use in the scope of the tasks they needed to perform.



Novo Stock > CTH01M100971

Ano Civil: 2012

Peso Paciente:

Nome Comercial	Nome FCC (Ultimo Ano)	Quantidade Consumida	Unidade de Medida
ADVATE	FVII		UI

Código	Génro	Data nascimento	Nat-País	Nat_Cidade	Resd-Cidade
CTH01M100971	F	10-09-1971	Portugal	Lisboa	Coimbra
CTH01M100971	F	10-09-1981	Portugal	Aveiro	Lisboa

Com um Click numa porção do Gráfico, poderá ver detalhes desses pacientes em forato tabelar

Fig. 3. Second version of the *Prototype-hemo@record*: same examples of the mockups (in Portuguese)

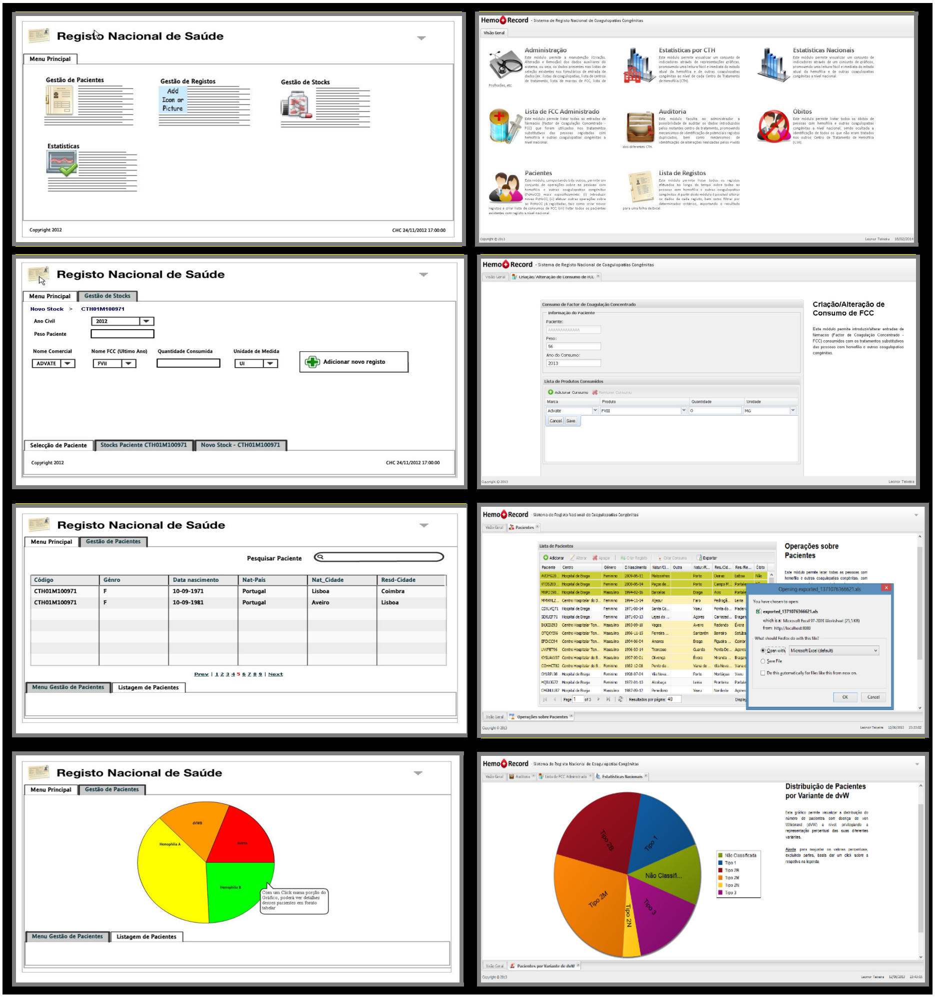
Fifth Step (2nd Prototype Validation and Reformulation). This version was iterated until the users were satisfied with the system and lasted about two weeks. Although this version does not add new functionalities, it allowed to analyze each of them with greater detail, since the tasks where unfolded to their most atomic elements. Moreover, this release was determinant in the selection of the type of system to implement (modular design), as the aspects related to the non-functional requirements, and more specifically the structure, organization and navigation, were assessed. Thus, a modular design was the selected approach, with each module responsible for implementing a single information process (workflow).

3.3 Results: Using Mockups and Prototyping to Identify System Requirements

After completing the process of requirements elicitation, performed iteratively and incrementally around the five phases previously presented, the analyst converted all the functional requirements identified in a formal representation, in order to facilitate the communication with the system programmers. For this representation, the UML notation was used, specifically the use-case diagrams to represent the actors and interactions of these with the application, and the class diagram to represent the conceptual model of the application domain.

Left Side: Examples of Mockups representations resulting from 2nd Prototype

Right Side: Example of User Interfaces representations resulting from the final application



Código	Génro	Data nascimento	Nat-País	Nat_Cidade	Resd-Cidade
CTH01M100971	F	10-09-1971	Portugal	Lisboa	Coimbra
CTH01M100971	F	10-09-1981	Portugal	Aveiro	Lisboa

Fig. 4. Mockups of the 2nd prototype and corresponding user-interfaces from the final application

Additionally, and as mentioned before, this process of requirements elicitation has also had a strong influence on the identification of some non-functional requirements, having the 2nd prototype a pivotal role in this issue. As such, 2nd prototype was thoroughly analyzed by programmers in order to develop the interface technology solution according to the mockups validated by users abreast of the functionalities identified in the use-case diagrams, and data model in a class diagram.

In a nutshell, Table 1 summarizes the main activities of the present RE process, the stakeholders involved in each activity, as well as the main inputs and outputs.

Table 1. Main activities, inputs and outputs of the RE

Activities	Start-up meeting	1st Prototype	2nd Prototype
Stakeholders	– Programmers – Systems analyst – Users /Clients	– Systems analyst – Users /Clients	– Programmers – Systems analyst – Users /Clients
Times / Period of evaluation	– One meeting	– About three months	– About two weeks
Inputs	– An idea and a requirement	– Outputs of the start-up meeting; – results of a previous study of the authors [14] – Results of an analysis of the national reporting systems already implemented in other countries	– Outputs of prototype 1 – Experience of the programmers
Outputs (main results)	– Objectives of the system – Some very high-level requirements	– High-level functional requirements – Some low-level requirements	– Low-level requirements – Non-functional requirements

4 Conclusions

It is an established fact that shortcomings in requirements elicitation can lead to inadequate implementations, thus leading to higher costs in the development of any software [15]. Particularly in the phase of requirements elicitation (one of the main critical stages of RE), the participation and collaboration of the stakeholders are crucial activities in the survey process and requirements elicitation. Beyond being a complex process, it can still become even more difficult if the stakeholders involved are geographically dispersed.

In reality, and particularly in the areas further away from engineering and computer science, the largest part of the requirements is in the implicit knowledge and experience of potential users, being difficult to extract this knowledge without an active and collaborative involvement of those potential users (clients/users). Accordingly, techniques that promote the requirements elicitation, and at the same time assist in converting the implicit knowledge (user experience) into explicit knowledge (documented knowledge) should be used.

In this article, and working in the domain of a complex health problem, the authors proposed a method for eliciting requirements based on participation, collaboration and negotiation of requirements by the different stakeholders of the project, supported by the method of prototyping. This method promotes an approach of stepwise refinement of requirements, starting with general ideas until the low-level requirements are achieved. Since this is a prototype, this method was also essential in the identification of some non-functional requirements that complemented the definition of the functional requirements.

This process involved a group of 10 members (8 of them physicians and future users), who started by defining the general objectives in a face-to-face meeting. After this meeting, the analyst designed the first version of the solution using Lumzy, having thereafter sent individual invitations to other participants with a small user manual. Each participant had access to the Web platform (using a login and password), and was able to define new requirements, modify existing and eliminate non-relevant ones by defining the interface required for each requirement. At the same time the participants used and modified the interface according to their needs, the analyst was notified of changes and collected the requirements. Several versions of the prototyping solution were achieved during this process, and the requirements were collected from a very broad and geographically distributed community of physicians, without the need to promote other joint meetings. In addition to the gathering of requirements, the platform allowed to drill-down into each requirement in terms of more elementary tasks, allowing the definition of the data model with full accuracy by the analyst.

It should be noted that in this study the rate of participation in the definition and validation of the requirements through this platform exceeded our expectations and the results were very satisfactory. Based on the experience obtained with this case, it is possible to conclude that prototyping tools like Lumzy are very appropriate to use in the process of requirements gathering, when the system involved is complex, and dynamic in terms of its definition, and the stakeholders of the project are in dispersed locations.

The results of the case study were promising, as they showed that for the example at hand, the method of requirements elicitation by prototyping can be crucial to the success of the software development project.

Acknowledgments. This work is funded by National Funds through FCT - Foundation for Science and Technology, in the context of the project PEst-OE/EEI/UI0127/2014. We would also like to acknowledge the valuable contribution of the clinical professionals of Portuguese Association of Congenital Coagulopathies (PACC) for all the help in the requirements analysis.

References

1. Laporti, V., Borges, M.R.S., Braganholo, V.: Athena: A collaborative approach to requirements elicitation. Computers in Industry 60(6), 367–380 (2009)
2. Teixeira, L., Ferreira, C., Santos, B.S.: User-centered requirements engineering in health information systems: A study in the hemophilia field. Computer Methods and Programs in Biomedicine 106(3), 160–174 (2012)

3. Teixeira, L., Ferreira, C., Santos, B.S.: Using Task Analysis to Improve the Requirements Elicitation in Health Information System. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS (2007)
4. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998, pp. 1–40 (1998)
5. Abran, A., et al.: SWEBOK - Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, California (2004)
6. Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, Heidelberg (2010)
7. Meth, H., Brhel, M., Maedche, A.: The state of the art in automated requirements elicitation. Information and Software Technology 55(10), 1695–1709 (2010)
8. Sommerville, I.: Software Engineering, 8th edn. Addison-Wesley, Harlow (2007)
9. Zhang, X., et al.: Applying evolutionary prototyping model for eliciting system requirement of meat traceability at agribusiness level. Food Control 21(11), 1556–1562 (2010)
10. Fu, J., Bastani, F.B., Yen, I.-L.: Model-Driven Prototyping Based Requirements Elicitation. In: Martell, C. (ed.) Monterey Workshop 2007. LNCS, vol. 5320, pp. 43–61. Springer, Heidelberg (2008)
11. Davis, A.M.: Operational prototyping: a new development approach. IEEE Software 9(5), 70–78 (1992)
12. Rudd, J., Stern, K., Isensee, S.: Low vs. high-fidelity prototyping debate. Interactions 3(1), 76–85 (1996)
13. Arnowitz, J., Arent, M., Berger, N.: Effective Prototyping for Software Makers, p. 624. Morgan Kaufmann Publishers Inc. (2006)
14. Teixeira, L., Ferreira, C., Santos, B.S., Savedra, V.: Web-enabled registry of inherited bleeding disorders in Portugal: conditions and perception of the patients. Haemophilia 18(1), 56–62 (2012)
15. Teixeira, L., Ferreira, C., Santos, B.S., Martins, N.: Modeling a Web-based Information System for managing clinical information in hemophilia care. In: Conference Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, pp. 2610–2613. IEEE Engineering in Medicine and Biology Society (2006)