Elicitation of Non Functional Requirement With Quality Policies

Zair Abdelouahab & Francisco bastos
Federal University of Maranhão, Electrical Engineering Department,
Av. dos Portugueses, S/N, CEP 65085-580,
São Luís MA, Brasil
(zair,bastos)@dee.ufma.br

Abstract

The present paper describes a method for non functional requirements elicitation in agreement with standardizing and quality policies for medical software, being its main goal to guarantee the consonance of such policies with the non functional requirements, when adopting new technologies and requirements for the existing software, as well as to define models to create those policies and to represent the non functional requirements, besides encouraging the creation of such policies in whatever institutions in lack of it..

1. Introduction

The increasing search for quality software pushes the market to attend not only the required functionalities, but also the non functional ones (NFRs) wanted by the clients. These requirements, as they are difficult to address and validate, are informally and secondarily called in software development processes with respect to its elicitation. We believe that by treating the non functional requirements just from its elicitation phase, we will be essentially contributing to the overall quality of the software to be produced.

In addition, the quality of medical assistance, influenced by the information technology advancements applied to the area, increases the need for high quality software [5]. Yet, different types of medical software have different quality characteristics, which must be considered when developing them. Changes in those requirements ought to be administered regarding minimum impacts on the software. The use of proper elicitation techniques can significantly contribute to enhance the quality of the software to be produced. This work has as main goals: to elicit non functional requirements for medical software, guaranteeing the agreement of those requirements with standardization and quality policies, whenever new requirements or new technologies were to be adopted.

2. The Proposed Method

The method is based in the following steps:

1. **Non Functional Requirements Elicitation Step:** Elicitation of non functional requirements (NFRs) must be performed by the requirement engineer, starting from the scenarios obtained during the elicitation of the system functional requirements. The method proposes that the requirement engineer must conduct an analysis in each scenario, as to find out which of the non functional requirements can be appl ed over the scenario under study. Each NFR that is found must be recorded in the List of Elicited non Functional Requirements (LENFR) (code NFR, Name, Definition, scenarios).

2. **Non Functional Requirements Decomposition Step:** The initially elicited NFRs are goals called *Primary NFR,* which must be satisfied*;* hence they must be decomposed into sub-goals called *Secondary NFR,* until the operations corresponding to actions necessary to meet the main goal are found.

3. **Interdependencies Identification Step:** In this step are identified the interdependencies positive and/or negatives among the NFRs. The degree of difficulty in identifying possible interdependencies with other NFRs is proportional to the requirement engineer's knowledge on the matter domain [3]. For this activity the method uses the *pair comparison* heuristics proposed by [3], in order to guarantee that all NFRs will be evaluated considering their inter-influence.

4. Non Functional Requirements Representation

Step**:** After elicitation, the NFRs need to be represented someway. This representation must allow us to deal with them orderly easing their handling. One way of representing the NFRs are the graphs proposed by [1] [2]; another way to represent them is the scheme addressed strategy.

5. **Policies Creation Step:** This step must be executed only when the policies were not created. To produce the policies, the method used the following activities: ***Create Standardizing Policy;*** *Vocabulary* ***Standardizing***; *Register; Standardizing Interfaces (User Interface, Services Interfac);, Messages Standardizing. To produce a quality policy we used the following steps:* ***Create Quality Policy;*** *Usability Policy, Reliability Policy*; *Maintenance Policy*,

6. **Conformity Evaluation Step:** This step is important to guarantee conformity between the non functional requirements specification and the existing policies. We use the following activities: ***Policy Evaluation:*** this activity uses the strategy based upon HoQ [4], in which as the policies are examined their statements are extracted and fulfilled the column of the equivalent policies table for further conformity evaluation; ***Conformity and Contradiction Identification:*** After fulfilling the Table, in the previous activity, this activity is started establishing relationship between requirements and policies, marking the cooperative relationships with the symbol '□' and the conflict ones with '□'. Therefore, the Table is examined and fulfilled with the found relationships.
Developing of an action strategy: The execution of this activity depends on the type of relationship found in the Table and can follow one out of two routes: return to the non functional requirements specification process, whenever a NFR is in conflict or is redundant with the existing policies; otherwise update the policies whenever they are not in conformity with the specified NFRs..

3. Conclusion

This work describes a method for non functional requirements elicitation in agreement with quality and standardization policies for medical software. The main focus was to guarantee its conformity with conformity with the policies, whenever new technologies or functionalities for software are acquired. Thus, as long as new technologies and functionalities are adopted, the elicited NFRs should agree with them. This method comprises six (6) steps which were applied to a case study to elicit non functional requirements of a hospital information system.

The proposed approach suggests a set of lists to help the system analyst during the requirement process, thus allowing the creation of quality and standardization policies through the policies creation step, in case they do not exist.

4. References

[1] Chung, L., Nixon, B. "Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach" Proc. 17th Int. Con.on Software Eng. Seatle, Washington, April pp: 24-28, 1995.

[2] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. "Non-Functional Requirements in Software Engineering" Kluwer Academic Publishers 1999.

[3] Cysneiros, L. M.; Leite, J.C.S. P. Requisitos não funcionais: Da Elicitação ao Modelo Conceitual. Tese de Doutorado submetida na PUC-Rio de Janeiro, em fevereiro de 2001.

[4] Hauser, R. J.; Clausing, D. The House of Quality. Harvard Business Review, 32(5), pp. 63-73,1988.

[5] Regina, Ana et all. Qualidade de Software Médico, UFRJ. 2000.