Approach to Define a Non-Functional Requirements Elicitation Guide Using a Customer Language

Andreia Silva, Placido Pinheiro, and Adriano Albuquerque
Graduate Program in Applied Informatics
University of Fortaleza (UNIFOR)
Fortaleza, Brazil
andrearsp@gmail.com, {placido,adrianoba}@unifor.br

Jonatas Barroso
Computer Department
Federal University of Ceara (UFC)
Fortaleza, Brazil
jonatascbarroso@gmail.com

***Abstract*—Non-functional requirements (NFR) have a crucial role in the software development process because they correspond to the characteristics and restrictions on which the software must running and represent factors that influence the time and cost of software development. Nevertheless, many organizations do not perform these requirements elicitation properly. This paper presents an approach to creating a NFR elicitation guide focused on customer language. To create the approach was performed a systematic review which identified and analyzed related works. Also a survey was conducted which helped to know the current situation of NFR elicitation in software development organizations and obtained suggestions for composition of the NFR elicitation guide. Finally, the results of applying the proposed approach presents evidence that the use of the process is feasible and produces better quality requirements.**

Keywords-process; approach; elicitation; non-functional requirements; NFR; customer

I. Introduction

All systems have non-functional requirements (NFR) but are not always explicitly defined in a formal specification [1]. NFR elicitation is a complex task and some factors contribute directly to this. Firstly, NFR are very diverse. This makes it difficult an analyst know all applicable kinds to a given software context. Customers, in turn, do not always know their software non-functional needs or do not know how to explicit them. In addition, the several knowledge sources, such as standards and norms or relevant bibliographic references related to NFR, do not specify in what situations a requirement should be elicited or define a pattern for its definition. Furthermore, while there have been proposed works dealing NFR elicitation, yet there is no evidence that indicate the most appropriate method for these requirements eliciting. None of these methods has been adopted as standard by the requirements engineering community [2].

II. METHODOLOGY

Conducting a systematic review of over 1700 publications identified related studies with the NFR elicitation. The main research question was identify approaches related with guides utilization to support the NFR specification. The search was conducted in digital libraries of ACM, IEEE and Scopus, including top conferences interested in requirements engineering studies. Then many papers that define procedures for NFR elicitation were analyzed and only two of these publications were considered related works.

Kopczynska and Nawrocki [3] present results of a case study on NFR elicitation through a method based on the knowledge reuse. This method consists of a short sessions sequence to discuss the ISO 25010 [4] characteristics. In each session are used predefined requirements with values to be replaced by analysts during the specification. New templates can be created from the NFR elicited and reused in other software projects. On the other hand, Balushi [5] propose a framework for NFR elicitation supported by quality requirements ontologies defined from the application context to be developed. The specification process is aided by a tool that contains a knowledge base of requirements already elicited. Each new project updates the database with new requirements identified.

In addition, a survey was held in order to get more information about the NFR elicitation in software development organizations. In summary, main objectives of the research were: (i) an overview of the current situation related to the practice of NFR elicitation and (ii) identify the reasons for these requirements are not elicited. This survey counted with the participation of 100 professionals representing several roles involved in software development. About 69.4% of respondents said elicit NFR in their companies. However, only 11.8% of these professionals consider that the NFR definition is good. Participants also reported the reasons why these requirements are not elicited often. The main reasons for this failure definition are: lack of knowledge of the teams, lack of request of these requirements by the customer, and lack of technical or financial organization capacity to comply with NFR. The survey also sought to evaluate the need for a guide to support the NFR elicitation. About 90% of participants said that a guide would be useful to identify non-functional requirements and half of all respondents reported suggestions for the contents of this guide.

III. Approach to Creating an Elicitation Guide

A literature systematic review provided an understanding of the main approaches related to the NFR elicitation. The survey, in turn, allowed to identify the expectations of the involved professionals in the software development process about what

DOI reference number: 10.18293/SEKE2016-195

should be addressed in the approach proposed in this paper. Thus, these steps contributed to define the proposed process. This approach is performed before the beginning of any organization's project and aims to create a NFR knowledge database that can be reused in projects. Thus, it translates the expert knowledge in the most appropriate questions and requirements to a better understanding of customers and business analysts, allowing disseminate this knowledge with other organization members. A preliminary version of this approach can be seen in [6]. This paper presents the evolved version, including details of how to perform each of the proposed activities and the results of user experience.

A. Approach Description

Fig. 1 shows an approach overview detailed in the following subtopics.

1) Define Software Type: The first process activity involves defining the software type for which you want to create the elicitation guide. Examples of software type classification can be found in [7] and [8]. To perform the choice of the corresponding type, the organization can analyze the developed products and seek to identify applications that require more attention in terms of NFR or simply select the application type that develops more frequently.

2) Select Reference Basis: This activity involves choosing a bibliographic source to guide the definition because it is not always easy to distinguish functional and non-functional requirements. Also, knowing many NFR types helps to define which type is important for the software that the organization develops. For that, some important references are available in the literature. Among the most relevant NFR catalogs is the list provided by Chung [9] which contains more than one hundred requirements types. Another requirements types classification is presented in ISO 25010 [4] standard and Sommerville [10].

3) Identify Experts: This activity consisting in identifying the organization professionals working in technical areas related to the NFR types. In some cases, the organization knows the most important requirements types for its context, in other cases may not know anything about these requirements. Thus, the responsible to implementation approach can make a prior analysis of requirement types addressed in reference database and identify experts available in the organization. These professionals will participate in this process activity until choose the requirements to be treated.

4) Select Requirements Types: After defining the reference database that will guide the requirements definition process and identifying the organization professionals who participate of this process, you must select the requirements types that will be part of work scope. The applicability of each requirement type for the software defined context should be evaluated in this step of approach.

5) Identify Questions: The next step involves analyzing each NFR type and verifying how the customer can be asked about the software requirements. This is one of the most important process activities, because it seeks to obtain a complex information in a simple and comprehensible to a user without technical profile way.

6) Define Requirements: Then, for each identified question should define requirements models, establishing clearly the customer needs and allowing the analyst to get the parameters for the elicited requirement. The attributes that must be defined for each requirement are shown in Table I.

7) Validate With Experts: This validation aims to verify if the requirements are understandable and relevant to the technical areas. In general, technical professionals need the information contained in the NFR to perform their jobs. Therefore, ensure that defined requirements meet this purpose is important.

8) Validate With Users: Then, the requirements must be evaluated by users; in this case anyone who has contact with the customer in the requirements elicitation activity. This evaluation aims to ensure that the requirements are suitable for the customer language, unlike the previous evaluation, which verifies that the requirements are feasible from a technical point of view.

TABLE I. GUIDE REQUIREMENTS ATTRIBUTES

Attribute	**Description**
ID	Unique requirement identifier in the guide.
Requirement Type	Requirement class to which the NFR belongs.
Concepts	Requirement type definition, described preferably based on norms, standards, technical articles, books or other recognized references.
Question	Question asked to the customer about the corresponding requirement. Responses should be simple and direct to avoid subjective interpretation.
Template	Requirement writing pattern, including the parameters. There are two parameter types: mandatory, delimited by < and >, and optional enclosed in [and].
Example	Requirement based on the established model including real values for the variables. We suggest that the example values are defined based on the organizational service capacity, because, this way, business analysts have the information than can be met by the organization at the time of negotiation with customers.
Mandatory	Indicates if the requirement is required or desirable.
Dependency	Indicates which other requirements should be defined.

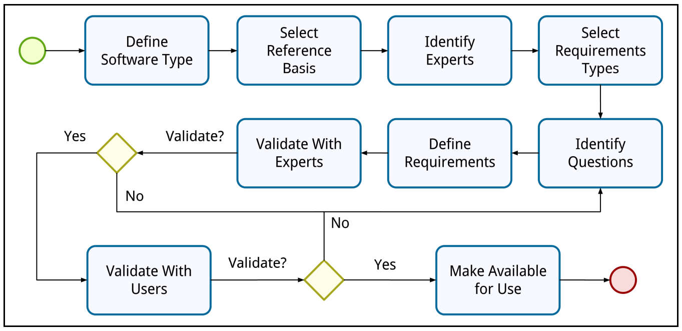

Figure 1. Process for Creation of the Elicitation Guide

9) Make Available for Use: Finally, after evaluation activities, the bidding guide should be available for use during the software requirements identification stages. The professionals involved in the requirements elicitation process must be notified of guide availability. Note that the guide can be used as a support tool in any requirements elicitation process.

IV. Approach Use Experience

In order to analyze the proposed approach feasibility, was held an use experience in an organization that provides IT and communication solutions for the Brazilian government. This organization has a defined process for requirements elicitation but there is no specific guideline to NFR. Approach application was conducted between July and September 2015.

A. Approach Execution

The following subsection describe how the execution of each activity that compose the approach was performed.

1) Define Software Type: During the discussions has been decided that web applications would be the software type to be treated. Although developing other applications types, most of organization customers requests involves this software type. After definition, managers requested to prioritize the requirements relating to security, performance and usability because, being a public organization, some standards published recently required the treatment of these requirements.

2) Select Reference Basis: During the activity execution, the NFR types sources were presented. Among them, the management team has selected as a reference the ISO 25010 [4] characteristics catalog. In addition to being an international standard, this catalog has a well-defined classification of requirement types that the organization had interest in dealing.

3) Identify Experts: For the experiment, the organization provided a group of professionals who perform activities in different technical areas. The choice of professionals occurred considering the performance of them in technical areas related to NFR. Experts were selected the areas of architecture, security, capacity and availability, and performance, among others. No less important, all selected professionals have over 8 years experience in IT and at least 3 years working in the organization. Regarding the academic level, 18% are masters, 45% specialists and 36% graduates.

4) Select Requirements Types: Choosing requirements types occurred by analyzing each subcharacteristic presented by ISO 25010 [4]. During this activity, experts evaluated each requirement type from the perspective of importance to the organization context who work and software selected type. No subcharacteristic related to Functional Suitability and Maintainability were selected for treatment. According to experts, the organization has strict contracts that do not allow variations in the compliance degree the defined scope for the project, unless there is change negotiations this scope. For this reason, the Functional Suitability characteristic was not selected. Regarding Maintainability, experts reported that the organization works with its own development framework that already includes a lot of good practices that seek to ensure a good maintainability level and therefore not need to establish such requirements. During the discussions, the team identified non-functional requirements that should be elicited but that did not fit into any of the standard characteristic. Then, three new subcharacteristics related to Technological Constraints and Legality have been added to the identified types list. Although not addressed by ISO 25010 [4], are cited by Sommerville [10] and were included in the scope of work because of the organization needs.

5) Identify Questions: After selecting the requirements types, each subcharacteristic was analyzed from the perspective of how the customer could be asked if the corresponding requirement would apply to the software. The proposed questions were evaluated rigorously to ensure that only "Yes" and "No" answers were possible. This was done to maintain a standard and ensure simple and direct questions. These questions aimed at capturing the requirement necessity. Other information required for the requirement are treated as attributes in the next step.

6) Define Requirements: This step was conducted with the expert support who were consulted to answer questions concerning the requirements applicability and about the organization indicators. Obtained values were needed to set requirement examples that should be defined based on organizational service capacity. Moreover, in certain cases, the organization does not keep important information concerning the capacity. Therefore, requirement definition that depend on this information is a motivating factor for the organization understand better its performance. For example, to define recoverability software requirements after a failure, the organization must know the time when the services were restored. This is not always known and the requirement definition alerts the organization to this fact. In addition, for each question at least one NFR model was suggested such that, when the customer answering it, the business analyst has predefined requirements to be elicited. Thus, 36 questions and 49 non-functional requirements have been defined. Table II shows NFR performance examples obtained with the implementation of the approach.

7) Validate With Experts: After requirements defining, each was analyzed using some criteria (e.g., completeness, clarity, relevance, and capacity to implement and test) and the comments were recorded and treated. This activity repeated until all requirements groups were fully validated.

8) Validate With Users: We selected three business analysts for this stage. All received, in addition to the elicitation guide, a brief orientation about the work purpose and use of the guide. Then users evaluated the guide and requirements according understandability and usability criteria. The evaluation result was satisfactory because only five comments were made about the clarity of few requirements. Thus, some improvements were made after the evaluation. However, it was noticed that the basic understanding of some technical issues are needed and training on these issues is essential to avoid possible doubts on the customers questions. Finally, all analysts said the guide is easy to learn and use. One user also reported that the guide would greatly help for the work performed by them.

9) *Make Available for Use:* After validation of experts and users, the NFR elicitation guide was made available to support the organizational process. An email informing about the guide was sent to all business and systems analysts.

V. Conclusion

Some related work have mechanisms to support the NFR elicitation. The proposed approach differs from these work because adds some points that are not covered by other studies. It is a process for a elicitation guide creation performed before the start of any project. With this, the organization has the ability to analyze in detail the technical issues relevant to the context of software developed by it. Thus, the customer will be asked about relevant points for the architecture, infrastructure and other key issues for software development from the early stages of the project. In addition, the approach defines requirement examples with parameters filled based on the organizational service capacity. Also provides the evaluation of the requirements quality, not only from a technical perspective, but also from the point of view to the customer language adequacy.

In addition, the approach presents concepts of each requirement type in order to facilitate the understanding of the business analysts. Therefore, along with the issues focused on the customer, it becomes an easy guide to learn and use in the requirements elicitation process. Once the guide is constructed outside the projects scope, enables optimizing the experts time who will not have to attend during the elicitation process of each developed project in the organization. In addition, an organization that does not have expert professionals in all NFR types can count with support of a consulting for the approach implementation and then enjoy the results generated, i.e., reusing this knowledge database to support elicitation of all your projects.

The result obtained using the approach was compared with elicited requirements without the use of the approach in other organization projects. It could be observed that the proposed approach contributed significantly to the NFR elicitation in terms of quantity, quality and diversity of the requirements.

The proposed approach has as main innovation integrate the customer in NFR definition process in order to promote a change of the existing culture currently, where it is believed that many requirements types cannot be obtained from the customer. To deal with this issue, the approach works the translation of specialized knowledge in questions and more appropriate requirements for customers and business analysts. In addition to allowing disseminate the knowledge of experts with other organization members.

TABLE II. Requirements examples using the approach

Type	Time Behavior
Definition	Degree to which a product or system performs its functionalities according to the specified response time, processing time and throughput.
Question	There is some functionality that are necessary to determine the response time?
ID	4.1.1.1
Template	The system should have the following limits of response times when exposed to load defined by Requirement 4.1.2.1, which deals with the number of transactions: • <operation>: up to <response time> in <percentage of acceptance> of the requests.
Example	The system should have the following limits of response times when exposed to load defined by Requirement 4.1.2.1, which deals with the number of transactions: • Digital document upload: up to 2s in 90% of the requests.
Mandatory	Yes
Dependency	4.1.2.1 become mandatory.

Application of the proposed approach has identified some improvement opportunities and a third approach version is being built including other factors, such as identifying ways of measuring the size of these requirements.

References

[1] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," proceedings of the 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE), San Francisco, CA, pp. 9-16, 2013.

[2] M. M. Rahman and S. Ripon, "Elicitation and modeling non-functional requirements - a pos case study," proceedings of the International Journal of Future Computer and Communication, vol. 2, pp. 485-489, 2013.

[3] S. Kopczyńska and J. Nawrocki, "Using non-functional requirements templates for elicitation: A case study," proceedings of the 2014 IEEE 4th International Workshop on Requirements Patterns (RePa), Karlskrona, pp. 47-54, 2014.

[4] ISO/IEC 25010:2011, Systems and software engineering/Systems and software Quality Requirements and Evaluation (SQuaRE)/System and software quality models, 2011.

[5] T. H. Al Balushi, P. R. F. Sampaio and P. Loucopoulos, "Eliciting and prioritizing quality requirements supported by ontologies: a case study using the elicito framework and tool," in Expert Systems 30, no. 2, pp. 129-151, 2013.

[6] A. Silva, P. Pinheiro, A. Albuquerque and J. Barroso, "A Process for Creating the Elicitation Guide of Non-functional Requirements," proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), vol. 2, pp. 293-302, 2016.

[7] D. Mairiza, D. Zowghi and N. Nurmuliani, "An investigation into the notion of non-functional requirements," proceedings of the 2010 ACM Symposium on Applied Computing, pp. 311-317, 2010.

[8] A. C. Guerra and R. T. Colombo, "Information technology: software product quality," "Tecnologia da informação: qualidade de produto de software," Science and Technology Ministry, Brasilia, DF, 2009.

[9] L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, "Non-functional requirements in software engineering," proceedings of the International Series in Software Engineering, vol. 5, p. 476. Springer, Heidelberg, 1999.

[10] I. Sommerville, "Software Engineering," Addison-Wesley Longman Publishing Co., Boston, MA, 2006.