

## Determining Interdependencies among NFRs in Agile Environment to Reduce Conflicts

Jawad Ali\*, Muhammad Shoaib Shoukat, Muhammad Faisal

Department of Computing and Technology, Iqra University, Islamabad, Pakistan  
jawad.nsd@gmail.com, ranashoaibtech@gmail.com, faisal.khan16@hotmail.com

### Abstract

NFRs play a very key role in the development of high quality software. As NFRs are quality attributes of the software system, so it must be dealt properly and with care in the early phases of software development otherwise it will lead the system to failure or dismissal of quality requirement at later phases. NFRs have a very unique ability to conflict and contradict among themselves. And the main reason behind this ability of NFRs is interdependency, which should be found and dealt properly in order to avoid requirement dismissal. Agile methodologies only focus on functional requirements but neglects non-functional requirements. In this work, we identified and highlighted all the possible dependencies between different NFRs in agile environment. In addition we also proposed a specific framework to cope up with the identified dependencies between NFRs.

**Keywords:** Non-Functional Requirements, Agile Methodology

Received on 12 December 2017, accepted on 28 February 2018, published on 22 March 2018

Copyright © 2018 Author *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.22-3-2018.154384

### 1. Introduction

Requirement engineering (RE) is the process of meeting the user expectations for the new software or already used software and the constraints under which it is required to operate and developed. It is the first phase of the software development process and involves the defining, analyzing, refining, feasibility study, documentation, specification and maintenance of the requirements. Requirement engineering is very important part of the software engineering that provides the techniques and guidelines for developing quality software [1]. There are two types of requirements i.e. functional requirements and non-functional requirements. Functional requirements describe what a software should do and non-functional requirements describe how that functionality should be performed by the system [2]. Functional requirements deal with the functionality of the software and it is the input, output and function of the software. The functional requirements include reporting, description of data, interfaces of application etc. Non-functional requirements (also called quality attributes) deal with the criteria of the

determining the functional requirements. There are two main classifications of NFRs i.e. Run time NFRs and non-run time NFRs. Run time NFRs include performance, security, availability while non-run time NFRs include portability, reusability, testability, modifiability etc. [3]. Sometimes one NFRs conflicts the other NFR. For example, if we need availability then we may have to compromise on security so there must be good trade-off analysis to provide a better solution [4]. NFRs are quality attributes of the software. It must be considered and deal in a proper way. Because of inadequately dealing with it can lead a project to failure or requirement dismissal at later phases. It is necessary to deal with NFRs at early stages because it is very important for improvement and agility of the software project [5]. For the agile software development the requirements must be analyzed prior to the development either by the prioritization and prediction techniques. The reduction of false positive in the requirements capturing stage can improve the quality of the software [6]. It is very necessary to determine interdependency among various NFRs and deal with it properly, in order to develop quality software [7]. Agile methodology enhanced the process of software development method, and during the last few years it

became very popular [8]. The term „„agile requirements engineering““ is used to define the „„agile way““ of planning, executing and reasoning about requirements engineering activities [9]. For the success of any software development, one of the most critical and important activity is the analysis of Non-functional Requirements (NFRs). Agile improved the process of software requirements but in requirement engineering it still has some limitations, like minimal documentation, change of budget and time estimation, and neglecting NFRs. Agile lacks to provide methods for the management and elicitation of NFRs [8]. Identifying and classifying NFRs along with FRs in order to reduce conflicts among various NFRs and analysis of NFRs in Agile software development is our main focus in this research.

## 2. Background

In order to incorporate NFRs, many researchers used the NFR framework and goal driven approach. This approach was developed for modeling of both NFRs and FRs and their interdependencies. All requirements are considered to be goals, whereas NFRs are considered as softgoals. To visualize these goals the AND/OR trees can be produced, in which one is used for NFRs and one is for FRs, and the interdependencies among them are shown as + or – characters. [10]. In SPL, NFRs are modeled by using an automated approach known as Product Line UML-based approach. This approach considered performance requirements in all phases of the SPL. This method uses UML for considering NFRs. The main limitation of this approach is that it only considered performance requirements, not all NFRs. Software requirements are of two types, FRs and NFRs, FRs are defined clearly whereas NFRs are not defined clearly [11]. NFRs hold a conflicting nature and may affect the functionality of the software positively or negatively. So, it is very important to handle NFRs by using some tools and methodologies because of its subjective nature. For automatic optimization, a new methodology was developed for representation of NFR system functionalities and relationships, which is known as Softgoal Interdependency Rule set Graphs (SIRGs). This approach is capable of assigning weight, cost, and priority given to NFRs. As NFRs are not equally in nature and are different from one another but this approach treated all NFRs equally, which is the main drawback of this approach [12]. Large knowledge in regard to NFRs is required while dealing with NFRs for capturing of NFRs pattern to be reused [13]. Objective pattern, problem pattern, alternative pattern, and selection pattern are the four kinds of NFR visualization patterns that have been described. Taking NFRs into consideration is also becoming more important as FRs of the software because of the increasing complexity of software [14]. Usability, load, and security are some NFRs, which are more critical for a software system. NFRs play a very important role in the Big Data analytics and smart cities planning [19].

Visualization frameworks and methodologies are discussed in this section. The interaction of different NFRs can be viewed by software architects and designers are the main advantage of the visualization framework [15]. NFRs are not linked with FRs by the software development methodologies [16]. In conventional requirements engineering, NFRs are not properly defined and lack of consideration of NFRs in agile methodologies. Project management and requirement quality metrics are the additional NORMATIC objectives that are used to provide extensions for the improvement of visualization of NFRs and their potential solutions [17]. Project managers, architects, and software developers used an innovative tool that is developed by taking into consideration a well-known agile software development methodology and project management methodology is the strong point of this approach. In software development, the most important requirements combined are FRs along with NFRs [18]. Considering NFRs early in the development phases has of great benefits and will be less costly. Whereas considering during later phases will increase its complexity and cost. Getting quality functional software, which properly deal and taking NFRs into account are the main expectations of the customers and stakeholders. The most difficult areas in requirement engineering is to deal with NFRs and from the start of this decade these have been ignored where the rate of success of software increases by considering both FRs along with NFRs.

## 3 Proposed Framework

In this section a five step integrated approach is proposed after thorough literature survey. The steps are described by diagrams to have clear understating of the flow of information in each step. After a thorough literature we have proposed an approach for Identifying and classifying NFRs along with FRs in order to reduce conflicts. The pictorial representation of the proposed approach given in figure 1 and the detail description is given below:

### 3.1 Requirements elicitation

In the present age we are surrounded by different Softwares. These softwares become the essential part of our lives. These Softwares are developed accordingly, keeping the requirements of the user in consideration. Like all other industries the software industry also has to satisfy their customers. To achieve the objective, the software industry adopted different techniques used by software developers to achieve the customer satisfaction. The purpose of this model is to elicit the user requirements and then prioritize them for next phase of development to minimize the development errors. Domain analysis is also performed at this stage. In elicitation of requirements we decompose the requirements by means of two methods such as 1) by refining and 2) by prioritization. In our proposed solution

we introduce a technique to elicit the user requirements by using Kano questioner.

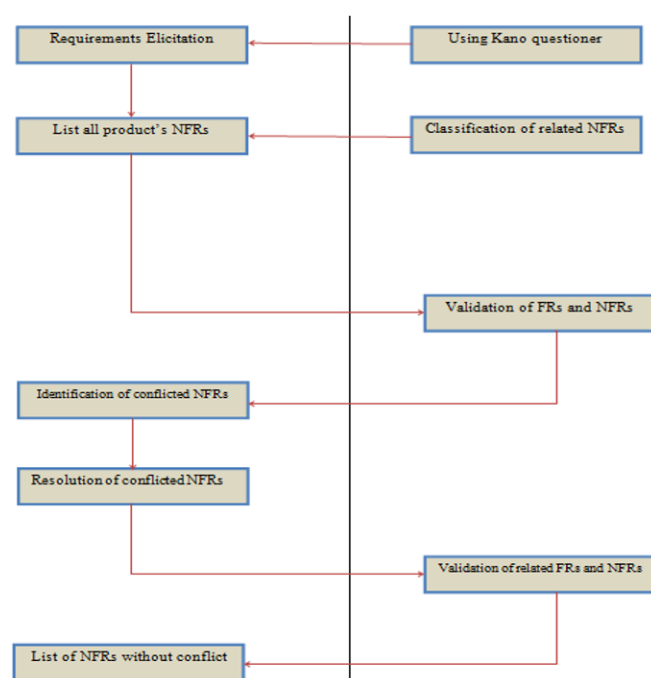


Figure 1. Proposed framework

The proposed method is divided into two components first is used for the elicitation of the user requirements and second component is used for prioritization of requirements. The first component is for eliciting the user requirements using a questioner. Before implementing 1st component, the stake holders involved are divided into classes and only the concerned requirements are evaluated by each stakeholder.

### 3.2 Kano Questioner

The 1st component questioner will contain two types of questions for each requirement/feature. The first question will ask about stakeholder's reaction if the product contains that feature, called functional question. And the second question will be about the feelings of stakeholder, if the product excludes that feature called dysfunctional question

### 3.3 Dealing with NFRs and their classification

As users/customers are not properly aware of software's non-functional requirements or quality attributes because they only focus on software's functional requirements. Customers are facing problems about the identification of NFRs. System developers are aware of the related technical details of the system. The developers point out

all the related NFRs of the project based on their knowledge after a clear validation. This step will help the customer's to overcome their lack of knowledge regarding NFRs and help them in the process of proper elicitation of NFRs [5]. For the ease of customers in proper identification of NFRs, the NFRs are classified into groups i.e. NFRs are classified on the basis of commonly used NFRs, conflicted NFRs, and meaning and attributes of NFRs. In which conflicted NFRs are further classified into absolute conflicted NFRs, relative conflicted and never conflicted NFRs. While NFRs on the basis of meaning and attributes are further classified into: with meaning and attribute, with only meaning, and with no meaning and attributes.

### 3.4 Dealing with Dependencies

Dependencies are the major cause of conflicts among various NFRs. It is necessary to keep track of the interdependence relations among various NFRs. In order to determine interdependencies among various NFRs, an approach known as NFRIF (NFR interdependency framework) is used in this step. The main strength of this technique is that it also considers FRs along with NFRs which is very helpful in proper designing of the system. NFRs are managed properly and chances of conflict are minimized by the use of NFRIF. This method is capable of dealing with seven different types of interdependencies.

### 3.5 Identification and resolution of conflicted NFRs

A mapping sheet is provided to the customer or user in this step in which each step is properly documented and provides ready to use template. By using predefined matrix, it also studies the FRs and NFRs relationship and conflicts [5]. According to their specification the NFRs are decomposed into internal and external quality attributes. It also determines interdependencies between these internal and external attributes and trade-offs among them. This process involves the user throughout the

process in order to validate the NFRs according to his required expectations. In order to produce NFRs, the working of both the customer and developer make it more reliable. The customer is often unaware of the project's related quality attributes, that's why these quality attributes are handled and decided by the system developers and project members internally. By the use of mapping sheet the developers will easily find out the conflicted areas and after identification and study of the conflicted situations, it is easily resolved and through this ready to use template or mapping sheet.

### 3.6 Validation from customer

After the using of all these techniques mentioned above, the collected and refined related project's requirements are then validated from the customer. After proper negotiation between both the customer and developer, that

they are agreed on the specified requirements or not. If both of them are agreed then a list of concern project requirements will be generated otherwise the whole process will be repeated again and again until both the customer and developer are agreed on the exact requirements. It is an iterative process due to which changes can be done easily.

### 3.7 Detailed diagram of proposed solution

Final and detailed diagram of the proposed solution is given below. This gives full detailed of the proposed framework from elicitation of requirements till generation of the final and a list of projects requirements without any conflicts.

## 4. Verification and Validation

This section discusses about the metric/ evaluation criteria for possible values of current approaches for the measurement in the first section. In the second section a case study of library management system is discussed. In the third section an online survey is performed in order to validate our approach and then analysis is made among the approaches based on possible values of the parameters. In the final section the proposed solution is discussed with detailed analysis.

### 4.1 Evaluation Criteria

Some evaluation criteria are needed for analyzing and evaluating the existing approaches and techniques. In order to compare the approaches the given criteria is defined. On the basis of evaluating the existing approaches as mentioned in the literature, these evaluation criteria are suggested. The techniques are evaluated based on the following parameters.

#### 4.1.1 Generalization

This approach is a general purpose approach and can be applicable for multiple platforms. It can be used for multiple requirements on multiple platforms.

#### 4.1.2 Methodology

A specific technique or method in order to elucidate the proposed approach is known as methodology. The method name and No are its values.

#### 4.1.3 User Involvement

In this proposed approach user is involved throughout the whole process. This gives the user full state of satisfaction during the whole process.

#### 4.1.4 Approach

To handle Non-functional requirements issues in a proper way is known as the approach. The name of the framework is described by the proposed approach.

#### 4.1.5 Notation/Language

This is used by the approach in order to argue their approach. This is also embedded in certain languages like UML, which uses UML diagrams etc. It is having the values of certain language if used or NO.

#### 4.1.6 Provide prototype

It provides a prototype which other approaches are unable to provide. Through this prototype customer can easily be satisfied and can ask for changes.

### 4.1.7 Conflict reduction

It means that this approach provides an environment in which the ratio of conflict is very low as compared to other approaches.

### 4.1.8 Integrating FRs with NFRs

This process integrates both NFRs along with FRs in order to provide an efficient product.

Figure 2. Methodologies likely values

No.	Parameters	Description	Likely Values
01	Generalization	Focusing related NFRs	Yes/No
02	Methodology	Proposed approach presentation method	Method Name/No
03	User involvement	User is involved throughout the development process	Yes/No
04	Approach	Approach proposed in the existing techniques	Approach Name/No
05	Notation/Language	Notation or language used by the existing approach	Language or Notation name/No
06	Conflict reduction	Minimizes conflict among NFRs	Yes/No
07	Integrating NFRs and FRs	Integrating both NFRs with FRs.	Yes/No
08	Using Prototype	Provides prototype for the ease and proper involvement of the customer.	Yes/No

## 5. Conclusion

In order to cover more aspects and keeping in view these pros and cons of each approach we proposed an approach for Identifying and classifying NFRs along with FRs in order to reduce conflicts. The proposed approach in summarized form is here as: Requirements are captured, analyzed and refined in the first stage. Based on refined requirements the constraints and dependencies are resolved through NFRIF. Classification of related NFRs is done in the form of prototype for the validation from customer. Integrating related FRs and NFRs. Related requirements are then validated from the customer/user. Identification and resolution of conflicts among various NFRs. Identifying the most critical and important NFRs. Then final list of related NFRs according to their importance is generated. The proposed approach provides the generalization, cost effective, reliability and parallel

processing, and can handle the complex requirements. The benefits of the proposed approach can be summarized as: It captures all the requirements because it proposed techniques in sequence by domain analysis, consulting stockholders and following best standards / practices. It provides iterative process for the ease of the customer to change and amend requirements at any stage during the development. This approach provides parallel processing which reduce time consumption in results. It provides a prototype for the user to validate requirements, which on the other hand provides reusability feature for the stakeholders. Keeps the customer involved throughout the development process.

## References

- [1] Daniel Melladoa, Eduardo Fernandez-Medina, and Mario Piattini "A common criteria based security requirements engineering process for the development of secure information systems", Science direct, 2007.
- [2] AmnaZeb, MahaChohan, MehreenSirshar, "A Review Analysis on Non-Functional Requirements and Causes of Failure of Projects", IJCCSE, 2015.
- [3] NidaAfreen, AsmaKhatoon and Mohd.Sadiq "A Taxonomy of Software's Nonfunctional Requirements", Springer, 2016.
- [4] MehrdadSaadatmand, Antonio Cicchetti, Mikael Sjödin"Model-Based Trade-off Analysis of Non-Functional Requirements: An Automated UML-Based Approach", International Journal of Advanced Computer Science, 2013.
- [5] S.Gayathri, S.Theetchenya "Requirements Engineering For Eliciting Non Functional Requirements Using Sprints In Agile Environment"Journal of Recent Research in Engineering and Technology 3(2), 2016, pp. 1-8.
- [6] Richard R. Maiti and Frank J. Mitropoulos "Capturing, Eliciting, Predicting and Prioritizing (CEPP) Non-FunctionalRequirements Metadata During the Early Stages of Agile Software Development" Proceedings of the IEEE Southeast Con 2015 - Fort Lauderdale, Florida, 2015.
- [7] MirzaRehenumaTabassum, Md. SaeedSiddik, Mohammad Shoyaib, Shah MostafaKhaled "Determining Interdependency Among NFRs to Reduce Conflict", 3rd International Conference On Informatics, Electronics & Vision, IEEE 2014.
- [8] BahiyaM.Aljallabi, Abdelhamid Mansour "Enhancement Approach for NonFunctional Requirements Analysis in Agile Environment", International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering, IEEE, 2015.
- [9] IrumInayat, SitiSalwahSalim, Sabrina Marczak, Maya Daneva,Shahaboddinamshirband "A systematic literature review on agile requirements engineeringpractices and challenges" Computers in Human Behavior (2014).
- [10] Q.L. Nguyen, "Non-functional requirements analysis modeling for software product lines", In Workshop on Modeling in Software Engineering (MISE '09), May, pp. 56- 61.
- [11] C. Burgess, A. Krishna, and L. Jiang, "Towards optimising non-functional requirements", In 9th International Conference on Quality Software (QSIC '09), Aug., pp. 269-277.
- [12] Richard R. Maiti and Frank J. Mitropoulos "Capturing, Eliciting, Predicting and Prioritizing (CEPP) Non-FunctionalRequirements Metadata During the Early Stages of Agile Software Development" Proceedings of the IEEE Southeast Con 2015 - Fort Lauderdale, Florida, 2015.
- [13] [S. Supakkul and L. Chung, "Visualizing non-functionalrequirements patterns" In 5th International Workshop onRequirements Engineering Visualization (REV), 2010, pp. 25-
- [14] B. Yin and Z. Jin, "Extending the problem framesapproach for capturing nonfunctional requirements", InIEEE/ACIS 11th International Conference on Computer andInformation Science (ICIS), 2012, pp. 432-437.
- [15] M. Umar and N.A. Khan, "Analyzing non-functionalrequirements (NFRs) for software development", In IEEE 2ndInternational Conference on Software Engineering andService Science (ICSESS), 2011, pp. 675-678.
- [16] W.M. Farid and F.J. Mitropoulos, "NORMATIC: Avisual tool for modeling nonfunctional requirements in agileprocesses", In IEEE Southeastcon, 2012, pp. 4-8.
- [17] M. Sadiq, "Modeling the Non-functional Requirements in the Contextof Usability
- [18] S. Supakkul and L. Chung, "Visualizing non-functionalrequirements patterns" In 5th International Workshop onRequirements Engineering Visualization (REV), 2010, pp. 25-34.
- [19] M Babar, F Arif, "Smart urban planning using Big Data analytics to contend with the interoperability in Internet of Things", Future Generation Computer Systems Vol. 77 pp. 65-76, 2017