Non-Functional Requirements Elicitation

GopichandMerugu

Abstract: Non-Functional requirements elicitation is one of the most important activity in requirements engineering. Non-functional requirements define the quality attributes of software system. If this analysis is not done properly, it may cause for problems in further phases that in turn may lead to failure of the software. Hence the non-functional requirements analysis has to be given at most priority in software development. This paper presents survey on different approaches proposed for Non-Functional requirements analysis.

Keywords : Non-functional requirements, Analysis.

I. INTRODUCTION

In requirements engineering, nonfunctional requirements (NFRs) plays an important role in the failure or success of software systems. Nonfunctional requirements are defined in several ways in the literature [1]. The definition for nonfunctional requirements is "a software requirement that describes not what the software will do, but how the software will do it" [2]. One more definition of NFRs is a description of property or characteristic that a software system must have exhibit or a constraint that it must respect other than an observable behavior of the system.

Thirteen NFRs are listed by the 'IEEE-Std 830 - 1993', which are supposed to be included in a software requirements specification document [3]. These

i. Performance requirements
ii. Resource requirements
iii. Documentation requirements
iv. Interface requirements
v. Operational requirements
vi. Safety requirements
vii. Quality requirements
viii. Security requirements
ix. Acceptance requirements
x. Portability requirements
xi. Reliability requirements
xii. Verification requirements
xiii. Maintainability requirements

So far NFRs are not treated as first class requirements by the majority of the researches. In fact, it is very difficult to incorporate nonfunctional requirements into the different life cycle phases of the software development. Many challenges are faced by researchers such as including large range of nonfunctional requirements, formally specifying requirements, subjective nature of nonfunctional requirements. These requirements can be incorporated into models. These models are used for resolving conflicts among NFRs and specifying functional requirements. Usually requirements are not independent but sometimes achieving one nonfunctional requirement may impact negatively achieving another nonfunctional requirement. For example, while ensuring security using multiple passwords (one nonfunctional requirement), the performance of the system (another nonfunctional requirement) may be affected because when it checks for multiple passwords, it takes more time, which in turn slows down the performance of the system. Most of the times, nonfunctional requirements are verified only after the implementation phase, i.e., from requirements phase to implementation phase, nonfunctional requirements are not considered explicitly and directly.

Due to this problem, there is a chance of getting requirements error at the later stages of software development. Some of the well-known problems in the development of a software because of not considering some nonfunctional requirements such as dissatisfaction of system users, system discontinuation and overruns of schedule and cost, are discussed in [4]. Nonfunctional requirements will have the impact on all phases of the software life cycle. The identification of NFRs should be complete, accurate and must be done as early as possible.

Generally, in industry, functional requirements are specified at the early stage of the software life cycle and nonfunctional requirements stated design level or at the implementation level [5,6,7]. If the nonfunctional requirements are not specified in analysis phase and specification phase, the product may get failed [6]. To overcome this problem NFRs must be considered in all the phases of the software development from the inception as per international standards. Various approaches are proposed like informal text, unstructured text and formal mathematical approaches [6]. The resources that are available and project goals are two factors to be considered in selecting an approach.

to the journal, rectification is not possible.

II.SEMIFORMAL AND INFORMAL APPROACHES

Unlike other approaches, semiformal and informal approaches do not need highly expertise persons which make these approaches at ease of use. These approaches cannot guarantee the completeness and ambiguity of the specifications obtained from system analysis. As the formal approaches are expensive and difficult to use, most of the people are used to prefer semiformal and informal approaches to specify nonfunctional requirements [8, 9].

The most popular work about nonfunctional requirements is the NFRs framework [2]. In the development process, NFRs framework considers nonfunctional requirements as soft goals which are supposed to be addressed.

Revised Version Manuscript Received on 10 September, 2019.
Gopichand.Merugu, Professsor& Head, Department of Information Technology, Vardhaman College of Engineering, Hyderabad, Telangana, India. (Email: Gopi_merugu@yahoo.com)

Goal graphs represent NFRs relations and their design decisions. In this graph the nodes represent design decisions or NFRs. Goals presented in NFRs framework can be elaborated into detailed concrete goals. This framework makes explicit relationships between intended decisions and NFRs. Each design decision may affect one or more NFRs negatively or positively which is better understood by the explicit relationship. The main purpose of this framework is that other models can reuse this framework to deal nonfunctional requirements. In [8] an approach is proposed which presents a semiformal model which is systematic and precise in nature. This model enhances the taxonomy of the nonfunctional requirements framework by combining the concept of the hard goals of NFRs in the requirements engineering process. This model provides the necessary information for early consideration of identifying, specifying, and separating functional and nonfunctional requirements

A new informal approach is proposed to discover requirements from stakeholder point of view [9]. Requirements of one stakeholder may conflict with requirements of other stakeholder, as many stakeholders may take part in system development. The pattern communicated by stakeholders is required to coordinate the requirements among the stakeholders. Nonfunctional requirements take a vital role as it influences most of the stakeholders concerns [9]. This approach is used to reduce the dissatisfaction of stakeholder and new possibilities can be found to satisfy other stakeholders. This procedure results improved specifications which are written in sequence diagram and the evaluation tables which contains the evaluation information of stakeholders. In the evaluation table the rows represent requirements types and columns represent the kinds of stakeholders. In this table a cell represents the type of requirement by the evaluation of stakeholders and three attributes are used to label each cell, these are reference of a refined specification, a score, and the content of the evaluation. This procedure is useful to validate the specifications completeness at the instance. This procedure does not work well when the requirement engineers do not interact with stakeholders in discovering requirements.

In [10], a semi-formal approach (CMU SEI's research result) is proposed which deals the impact of quality attributes on software architecture. The main focus of this approach is the terminology in taxonomy that can be used as a vocabulary to provide nonfunctional requirements and this vocabulary plays vital role in architectural design. The taxonomy is categorized into security, safety, performance, and dependability which are considered as various areas. Methods, concerns, and factors are the dimensions to analyze the quality attributes. Firstly, methods specify how to address the requirements. Secondly, the attributes of a system are presented by the requirements or concerns. Thirdly, system and its environment properties are considered as factors which might have effect/cause relationships and may impact requirements.

In [11], a semiformal approach is proposed for elicitation of nonfunctional requirements. This approach is also used for documenting the efficiency requirements. Quality models and quality attributes are used to capture the knowledge about nonfunctional requirements, some of which are provided in a template. This approach uses a quality model and generalized meta-model to capture the knowledge about other nonfunctional requirements.

In [12], a Four Layered approach is proposed for nonfunctional requirements analysis. As part of this approach some conceptual rules and metrics are proposed. Conceptual rules are used in the identification process and metrics is used to check the completeness of identified non-functional requirements. By this approach all possible Non-Functional requirements can be elicited.

In agile development [13], there are no explicit approaches for elicitation of nonfunctional requirements. In [14], an agile approach addresses the importance of performance requirements specification and testing which proposes a model called PREM (Performance Requirements Evolution Model). This model is used for identification and specification of performance requirements by development team. Initially these requirements are casual descriptions, and thereafter specifications are refined to desired level of detailed descriptions.

In [15, 16, 17], some approaches called misuse cases are proposed to deal with security requirements based on Unified Modeling Language (UML). Misuse cases are the descriptions of the set of sequence of actions which are not allowed by the system. In [15], a method is proposed based on misuse cases, which is used for deriving the nonfunctional requirements and functional requirements. Nonfunctional requirements analysis by means of threats, assets, counter measures and misusers is used to complement project and software requirements. In [17] an approach is proposed for differentiating both misuse cases and security use cases. These security use cases and misuse cases are used to specify and analyze the security requirements and security threats respectively. Both of these include the properties like the semantic inconsistencies of UML, inherit popularity and simplicity.

In [18], a Five layered approach is proposed for elicitation of performance requirements. As part of this approach some conceptual rules and metrics are proposed. Conceptual rules are used in the identification process and metrics is used to check the completeness of identified performance requirements. By this approach all possible performance requirements can be elicited.

In [19, 20] a semiformal approach is proposed, which is called Knowledge Acquisition in autOmated Specification (KAOS). It is used for elicitation of functional requirements and also used to analyze and model these requirements. This model is the combination of various sub models which are connected by means of consistency rules among these sub models with first-order temporal logic.

In [21] KAOS has been extended by Lam which handles security requirements. In this approach, for elicitation of security goals like integrity, confidentiality, availability, authentication, non-repudiation and privacy, few generic specification patterns are proposed. This approach contains two concurrent models which are incrementally built and specified to elaborate KAOS security requirements.

First one is an intentional model of the system-to-be and the second one is an intentional anti-model. This anti-model yields capabilities and vulnerabilities, which are used to achieve anti-goals of major security goals which are from the original model. This method constructs threat trees incrementally by means of refinements of anti-goals till the leaf nodes are reached. The leaf nodes are either implementable anti-requirements or observable attack vulnerabilities by the attacker.

III.FORMAL APPROACHES& RESULTS

Elaboration method of formal nonfunctional requirements is written in a formal language is popularly used. To specify requirements using formal methods is difficult and cost effective. As it is cost effective, formal methods can be applied only to critical nonfunctional requirements like security requirements. If the nonfunctional requirements are not specified properly, it leads to a huge loss with respect to money, data and time.

In [22] Formal Design Analysis Framework (FDAF) is proposed. This framework supports the systematic design of the software system which meets nonfunctional requirements like security, performance etc. To describe software architectures various notations have been used like formal methods and UML, which supports the design analysis [23]. Requirements engineers, formal methodologists and designers, are the stakeholders who develop the software system design by using this FDAF framework so that it has to meet nonfunctional and functional requirements. The requirements specification (functional requirements and nonfunctional requirements) and an object oriented design model are the inputs for this framework. This framework used to translate an extended semiformal UML design to a formal design, and also used to select the formal method. It is difficult to study various interactions among nonfunctional requirements using FDAF because it formalizes NFRs using various formal notations. Using this FDAF, it is also difficult to detect conflicts among various types of nonfunctional requirements.

In [24, 25, 26] some formal specification languages are proposed for component based software systems to specify non-functional properties. In [25] formal specifications of timeliness properties are proposed to describe the system by making use of temporal logic of actions in which states are represented as values which are assigned to state variables. In [24] a specification language called Component Quality Modeling Language (CQML) is proposed for Quality of Service properties of component based system.

In [27] a method for nonfunctional requirements formalization is proposed and a language called NoFun is presented. This language contains three parts, first one is the definition of software quality characteristics and attributes, in second one quality attributes of components are assigned with values, in third one both context-dependent and context-free quality requirements are specified over components. This language consists of type definition elements, structuring mechanisms. This language can also support to define non trivial quality models by other constructs. In [27], mapping of these concepts to Unified Modeling Language is studied by using some extension mechanisms. In [28] a method is proposed which describes the specification of formal requirements so that it is intended to increase the reliability and safety of the railway system by means of Z notation.

IV.CONCLUSION

The acceptance of any software product by the customer depends on how well we consider the Non-functional requirements. There is a great need for sophisticated methods and supporting tools in order to successfully carry out non-functional requirements engineering activities. In this paper several useful methods available for non-functional requirements analysis are presented. These methods include Four Layered approach for non-functional requirements identification, formal and semiformal approaches. Informal and semiformal approaches are simple to use.

REFERENCES

1. Lawrence Chung, Brian A. Nixon, Eric Yu "Nonfunctional Requirements in Software Engineering", Bostan, Kluwer Academic Publishers,2000.
2. Martin Glinz "On Non-Functional Requirements", 15th IEEE International conference on Requirements Engineering, Delhi October 2007, pp. 21-26.
3. IEEE Std 830-1993 IEEE Recommended Practice for Software Requirements Specification-Description.
4. Jaime De MeloSabatNeto, Julio Cesar SampaioLeite, and LuizMarcioCysneiros,"Non-Functional Requirements for Object-Oriented Modeling", In WER, 2000, pp.109-125.
5. Armin Eberlein and Julio Cesar Sampaio do Prado Leite, "Agile requirements definition: A view from requirements engineering", 2002.
6. AbderrahmanMatoussi and RegineLaleau, "A Survey of Non-Functional Requirements in Software Development Process", Technical Report TRLACL 2008, pp.1-15.
7. M.Gopichand, A.AnandaRao, K.Narender Reddy and J.Kiran Kumar, "An approach to requirements elicitation and analysis using goal" ICSTE 2010: 2ndIEEE international conference on Software Technology and Engineering, October 3-5,2010 San Juan, Puerto Rico,pp.218-221.
8. MohamadKassab, M. Daneva, and Olga Ormandjieva, "Scope Management of Non-Functional Requirements", In EUROMICRO '07: Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE Computer Society, 2007. Pp. 409-417.
9. Haruhiko KAIYA and Kenji KAIJIRI, "Refining Behavioral Specification for Satisfying Non-functional Requirements of Stakeholders", In IEICE transactions on information and systems Vol.E85-D, No.4(20020401), 1999, pp. 623-636.
10. Mario Barbacci, Mark H. Klein, Thomas A. Longstaff, and Charles B, Winstock, "Quality Attributes, Technical Report", CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University, 1995.
11. JrgDrr, D. Kerkow, Antje Von Knethen, and Barbara Paecha, "Eliciting Efficiency Requirements with Use Cases" In In: Salinesi C, Regnell B, Kamsties E (Hrsg) Proceedings of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), 2003.
12. Gopichand.Merugu, A.AnandaRao,"Four Layered Approach to Non-Functional Requirements Analysis" International Journal of Computer Science and Issues (IJCSI), ISSN: 1694-0814, Volume 8, Issue 6, pp 371-379, November 2011.

13. FraukePaetsch, Armin Eberlein, and Frank Maurer, "Requirements Engineering and Agile Software Development", In WETICE '03: Proceedings of the Twelfth International Workshop on Enabling Technologies, page 308. IEEE Computer Society, 2003.
14. Chih-Wei Ho, Michael J. Johnson, Laurie Williams, and E. Michael Maximilien, "On Agile Performance Requirements Specification and Testing", In AGILE '06: Proceedings of the conference on AGILE 2006, IEEE Computer Society, 2006, pp. 47-52.
15. Andrea Herrmann and Barbara Paech, "Quality Misuse", REFSQ -International Workshop on Requirements Engineering: Foundation for Software Quality, 2005.
16. GuttormSindre and L. Opdahl, "Eliciting Security Requirements with Misuse Cases", Requir. Eng.2005, pp. 34-44.
17. Donald Firesmith,"Security Use Cases", Journal of Object Technology,2(3), 2003, pp.53-64.
18. Gopichand.Merugu, A.AnandaRao" Five Layered model for identification of software performance requirements" International Journal of Software Engineering and Applications(IJSEA), ISSN: 0975-9018, Volume 3,No.5, pp 47-61,September 2012
19. Axel Van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour", In RE '01: Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE '01), page 249, Washington, DC, USA, 2001.IEEE Computer Society.
20. Emmanuel Letier, "Reasoning About Agents in Goal-Oriented Requirements Engineering", Ph.D. Thesis, ftp://ftp.info.ucl.ac.be/pub/thesis/letier.pdf, 2001.
21. Axel Van Lamsweerde "Elaborating Security Requirements by Construction of Intentional Anti-Models", In ICSE '04, Proceedings of the 26th International Conference on Software Engineering, pages 148-157, Washington, DC, USA, 2004. IEEE Computer Society.
22. Kendra Cooper, Lirong Dai, Yi Deng, and Jing Dong, "Developing a Formal Design Analysis Framework", In Software Engineering Research and Practice, 2003, pp. 68-73.
23. G. Booch, I. Jacobson, and J. Rumbaugh, "The Unified Modeling Language User Guide", Addison Wesley, 1998.
24. Jan OyvindAagedal, "Quality of Service Support in Development of Distributed Systems", PhD Thesis, University of Oslo, 2001.
25. Steffen Zschaler, "Formal Specification of Non-functional Properties of Component- Based Software", Workshop on Models for Non-functional Aspects of Component- Based Software (NfC'04) at UML conference 2004, September 2004, Technical Report UDFI04-12 Sept.2004 at TechnischeUniversit¨at Dresden.
26. Steffen Zschaler, "Towards a Semantic Framework for Non-functional Specifications of Component-Based Systems", Proc. EUROMICRO Conf. 2004, Rennes, France, September 2004, IEEE Computer Society.
27. Pere Botella, Xavier Burgues, Xavier Franch, Mario Huerta, and Guadalupe Salazar, "Modelling Non-Functional Requirements", Proceedings of Jornadas In-geniera de RequisitosAplicados (JIRA), Sevilla, Spain, 2001.
28. Miyoung Kang, Dae-Yon Hwang, Junkil Park, Jin-Young Choi, and Jong-Gju Hwang, "Formal Requirements Specification for Railway Signaling Systems", In ABZ 2008, London, UK, September 16-18, 2008.

AUTHORS PROFILE

GopiChand.Merugu received Ph.D. in Computer Science &Engineering from JNTUA, Anantapur, India and he received his M.Tech. in Software Engineering from the same university. He received B.Tech. degree in Information Science & Technology from Nagarjuna University, India. He is Professor & Head, department of Information Technology Vardhaman College of Engineering, Hyderabad. He is a member of IEEE, ACM, IAENG, CSI and ISTE