ISSN:0975-9646

A Survey to Support NFRs in Agile Software Development Process

Trupti Suryawanshi[#1], Gauri Rao[*2]

[#]*M.Tech , Computer Department, BVDUCOE, Pune, Maharashtra, India*

[*]*Associate Professor, Computer Department, BVDUCOE, Pune, Maharashtra, India*

***Abstract*— Software testing is the process of verification and validating the product. This process detects the difference between actual output and expected output. Agile software development methods promote adaptive planning, system development, timely delivery and scalability. There is various agile software development methods are available such as scrum, XP, Feature Driven Development (FDD) etc, used to deliver quality functional requirements (FR). In software development process software engineers are mainly concentrating only on functional requirements under the pressure of deploying the software as early as possible. But NFR is also important element of the development process. NFR is responsible for success and failure of the system. In this paper we have presented survey on need of generation of non functional requirements such as security, performance etc. NFR describes not actually what system will do but how the system will do it.**

***Keywords*—** Agile software process, FR, NFR, Requirement engineering**..**

I. INTRODUCTION

Now a day's agile software development methodologies (such as Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Agile Modelling(AM) etc.) are attracted towards the developing and delivering a quality software which may fulfil customers all requirements. But most of the developers are fail to notice the Non-functional requirements like performance, Scalability, usability, reliability etc.

Functional Requirement defines proper functionality of the software whereas a non functional requirement defines properties, qualities and constraints of the software.

Due to the short time developers only concentrated only on Functional Requirements. To develop any software NFRs are equally responsible as FRs. Lack of support of Non Functional Requirements in agile methodology may be responsible for failure of the software. Integrating of NFR with FR produces effective software.

II. BACKGROUND

A. Agile Development Process:

Agile development methods have been designed to solve the problem of delivering high quality software on time under constantly and rapidly changing requirements and business environments.

Features of agile software development method are as follows:

- Deliver quickly
- Simplicity
- Face to face communication between stakeholders
- Continuous attention to technical excellence an good design
- Self organizing team

Agile development software is allows for an adaptive process in which the team and developer react to and handle changes in requirements and specifications, even late in the development process.

B. Non Functional Requirements

In software engineering the term NFR is not related with the functionality of the software system. NFR does not have standard definition. Different authors defined different definitions of the NFR. Some of them are stated as below:

"NFR is defined as the software requirement which does not describe what actually software will do but how the software will do it. e.g. software quality attributes, performance, security etc."

"NFR is the non-behavioural requirements of a system, constraints and quality of software requirements."
"NFR defines overall qualities of the software system which restrict the product being developed and specify external constraints that the product must meet."

C. Types of NFRs:

Reliability, Performance, Security, Usability, Supportability, Efficiency, Verifiability, Interoperability, Maintainability, Flexibility, Portability etc.

Let us discuss how NFR is used for building any software system.

D. EXAMPLE: ATM MACHINE

Consider an example of developing software to operate ATM of any bank. System takes user transaction as an input (e.g. check available balance, deposit, withdraw etc.) and send that information to the central bank account system. After receiving an acknowledgement from central system transaction has been processed and responds back to user (e.g. display available balance, acknowledge deposit, dispense cash etc).

Some examples of NFR:

1) *Security Requirements:* System provides authorized access with valid bank card and personal identification number; record all transactions in daily

log, shutdown the system upon detecting device error, linking error or any software error etc.

2) *Performance requirements*: System shall respond to user within a limited time (less than 10 seconds).

3) *Restart requirement:* System shall automatically restart after failure of system within particular time (less than 10 minutes).

4) *Maintainability requirement*s: Mean time to repair is not more than 2 to 3 hours. Whereas Mean Time to Repair is the total time required to fault identification, correction and restoration of system for each failure divided by number of failures.

5) *Expandability requirements:* System shall designed in such a way that it can allow to future addition of system services as per market need.

From this example it is clearly understood that NFR is the integral part of software development process. Improper treatment with NFR is the main reason for software failure.

III. Related Work

Academic as well as industrial researchers attracted towards the non functional requirements. Most of the researchers agree that NFRs have been ignored in software development process and mainly in agile methodologies.

Pratima Singh and Anil Kumar Tripathi [6] clearly stated that mostly software engineers have concentrated only for testing FRs. So to produce high quality software consideration of NFR is very important. But practically managing NFR is not easy due to its flexible, subjective and delicate nature, and all the more due to great diversity, giving rise to conflicting requirements. Hence they suggest to NFR testing more effectively.

Abderrahman Matoussi and Regine Laleau [10] discuss a use of NFRs in software development process. In this paper they state that NFRs plays important role in success as well as in failure of the software. In software development process NFR is sensitive part.

Md. Mijanur Rahman and Shamim Ripon[4] states that proper management of the software requirements is very important for the further software development process within a limited time and cost.

As discussed above agile requirement engineering techniques ignores NFRs. But to design good quality software, developers must think about resources and quality attributes. Nupur Chugh and Aditya Dev Mishra[8] proposed assimilation of four layered approach to NFR which is helpful to improve quality attributes in software development process which would be used in the prioritization of user stories. This approach improves analysis of the requirements by considering NFRs that is essential part in software success.

IV. Issues And Challenges

To develop any software, requirement gathering and analysing are plays important role. FRs is easily collected from stakeholders but discovering NFRs is not easy task. There is no proper NFR generation method is available. Few methods have been proposed with some advantages and its disadvantages. But still there is not yet standard method defined by requirement engineering.

In the generation of NFR problem occurs due to the following aspects:

1) There is no clear idea about the NFR concepts. Different authors define different definitions but no commonly accepted definitions. Most of the definitions are inconsistent with each other. Generic and clear representation of NFR concept is needed.

2) Different views and understanding of NFR causes conflict that hard to be solved and prioritized.

3) Lack of supportive methods and tools to generate, specify and represent specific set of quality attributes of system.

4) NFR is not uniform in nature. NFR define global constraint on system hence NFR arises from all part of the system and interdependencies between them.

These are some common problems occurs during NFR generation. To overcome these problems common taxonomy, solving conflict, improving knowledge of stakeholder and additional knowledge supports are some common solutions.

From this survey we propose to develop a proper technique which easily produces NFRs. In our future work we design a tool which helps agile development team modelling NFRs early on during requirement gathering and analysis phases. This tool integrates FRs and NFRs modelling under one tool.

V. Conclusion

The success of any software depend on the quality of software whether the customer is satisfied with that software and its all requirements are fulfil by the software. Lack of support to NFR is responsible for the failure of system. To overcome this problem researchers face many challenges. This paper suggests for designing a tool to develop NFRs by considering that NFR plays sensitive role in the success of the software system.

Acknowledgment

I would like to express my special thanks to my guide Gauri Rao, who has guided me well in this, also would like to thanks my principal for giving me opportunity to work on this. Lastly I would like to thanks all peoples who are directly or indirectly involve making this trough.

REFERENCES

[1] Andrea De Lucia and Abdallah Qusef "Requirements Engineering in Agile Software Development", Journal of Emerging Technologies in Web Intelligence, Vol. 2, No 3, August 2010.

[2] Kaushal Pathak an Anju Saha "Review of Agile Software Development Methodologies", International Journal of Advannced Research in Computer Science an Software Engineering,Vol.3,Issue 2,February 2013.

[3] Malik Qasaimeh, Hossein Mehrfard, Abdelwahab Hamou-Lhadj, "Comparing Agile Software Processes Based on the Software Development Project Requirements", CIMCA 2008, IAWTIC 2008, and ISE 2008.

[4] Md. Mijanur Rahman and Shamim Ripon, "Elicitation and Modeling Non-Functional Requirements –A POS Case Study", December 20, 2012.

[5] Frauke Paetsch, Dr. Armin Eberlein, Dr. Frank Maurer, "Requirements Engineering and Agile Software Development".

[6] Pratima Singh and Anil Kumar Tripathi, "Issues In Testing of software with NFR", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.4, July 2012.

[7] David Ameller, Oriol Collell, Xavier Franch, "ArchiTech: Tool Support for NFR- Guided Architectural Decision-Making", June 25, 2012.

[8] Nupur Chugh and Aditya Dev Mishra, "Assimilation of Four Layered Approach to NFR in Agile Requirement Engineering", International Journal of Computer Applications (0975 – 8887) Volume 78 – No.5, September 2013.

[9] Gayane Azizyan, Miganoush Katrin Magarian, Mira Kajko-Mattson, "Survey of Agile Tool Usage and Needs", Agile Conference,,IEEE,2011.

[10] A. Matoussi, R. Laleau. "A Survey of Non-Functional Requirements in Software Development Process" October 2008.

[11] Chian Wen Too, Sa'adah Hassan, Jamilah Din, and Abdul Azim Abd. Ghani, "Towards Improving NFR Elicitation in Software Development", International Journal of Information Technology & Computer Science (IJITCS) (ISSN No : 2091-1610) Volume 7 : No : 1 : Issue on January / February, 2013.

[12] G. Goth, "Agile Tool Market Growing with the Philosophy", 2009, IEEE Software, 26(2), pp. 88-91

[13] G.Sousa, G.Silva, and J.Castro, "Adapting the NFR Framework to aspect Oriented Requirements Engineering", in the 17th Brazillian Symposium on Software Engineering (SBES), 2003, Manuas, Brazil.

[14] Jayakanth Seinivas and Kristina Lundqvist, "Using agile methods in software product development: A case study", Information Technology: New Generations, Third International Conference , vol.0,1415-1420,2009.

[15] A.Sidky and J.Arthur, "Determining the Applicability of Agile Practices to Mission and Life-Critical Systems",Proceedings of the 31st IEEE Software Engineering Workshop, IEEE Computer Society,2007,pp.