MANAGING NON-FUNCTIONAL REQUIREMENTS IN SCRUM

Presented by

Ms Saba

Registration Number: 358-FBAS/MSSE/F12

Supervisor

Ms. Zakia Jalil

Dr. Naveed Ikram

Department of Computer Science & Software Engineering

Faculty of Basic and Applied Sciences

International Islamic University, Islamabad

2017

SCRUM (computer Software development)
computer software – Development.

Department of Computer Science and Software Engineering

International Islamic University Islamabad

Date: [29-December-2017]

Final Approval

This is to certify that we have read the thesis submitted by **Saba, 358-FBAS/MSSE/F12** It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of **MS Software Engineering.**

Committee:

External Examiner:

Dr.Huma Hayat
Assistant Professor
National University of Modern Language[NUML]

Internal Examiner:

Dr. Tehmina Amjad
Assistant Professor
International Islmaic University Islamabad

Internal Supervisor:

[Ms.Zakia Jalil]
Assistant Professor
International Islamic university Islamabad

External Supervisor

Dr. Naveed Ikram
~~Associate~~ Professor
Riphah International University Islamabad.

Dedication

I dedicate my research work to the
best girls in my life
My Gorgeous Mother
My Beautiful Sisters
And to
My Lovely Daughter

Saba

358-FBAS/MSSE/F12

A dissertation Submitted To

Department of Computer Science and Software Engineering,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfilment of the Requirement for the Award of the

Degree of

MS Software Engineering (MSSE)

Declaration

I hereby declare that this Thesis "Managing Non-functional requirements In Scrum" neither as a whole nor as a part has been copied out from any source. It is further declared that I have done this research with the accompanied report entirely on the basis of my personal efforts, under the proficient guidance of my teachers especially my supervisor Ms. Zakia Jalil and Co-supervisor Dr. Naveed Ikram. If any part of the research is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, I shall stand by the consequences

Saba

358-FBAS/MSSE/F12

Acknowledgements

Praise and gratitude to Allah Almighty, the light of heavens and the earth, who guides our hearts to knowledge, reasoning and the truth. With His help and guidance everything is possible.

I feel my privilege to express my sincere gratitude to my supervisor Ms. Zakia Jalil and my external Supervisor Dr. Naveed Ikram, for all their kind help, guidance, suggestions and support during the research for this thesis. I am also grateful to my parents, husband and specially my sister Sheezah Taimouri who helped me a lot during my work.

I would also like to express my most sincere gratitude and thanks to my friends, class fellows and my special thanks to my teacher Ms. Saima Imtiaz who guided me throughout my thesis and helped me completing my research work.

Abstract

Scrum has been recognized as one of the most effective methods in delivering functional requirements of the system. At the same time, the researchers criticize it on neglect of non-functional requirements (NFRs). Mainly the negligence of NFRs is reported in requirement engineering (RE) phases of the software development lifecycle. It is stated as one of the major challenges that agile teams suffer in agile RE practices. However, Literature lacks the details on how NFRs are managed in the industry and thus fails to highlight the challenges scrum teams face while dealing with them. literature also reports number of consequences of the said issue for example major change request at the end of the project, over budgeted projects, late projects and sometimes major failures are reported due to missing some critical NFRs. In order to find out the challenges scrum industry is facing a survey study has been conducted with in Pakistan. Product owners of different scrum organizations were interviewed for in-depth analysis of the floating issue. Based on the survey results solution to the identified problems was designed in the form of guidelines. Proposed solution provides set checklist with supporting guidelines for all scrum ceremonies. Proposed guidelines managed to overcome all the major issues by eliminating the root causes of the issues. After solution design a focus group meeting of agile experts from the industry was conducted for validation purpose. With minor changes experts accepts and approved the guidelines. According to them proposed guidelines are easy to follow and implement and will provide long term benefits to the software industry.

Keywords *– Non-functional requirements, Agile, Scrum, software development lifecycle*

Contents

List of Figures

List of tables

Chapter 1
Introduction

1. INTRODUCTION

This chapter provides a brief introduction to the conducted research. It provides an overview of the research background, main terminologies and the problem statement; it also highlights research objective, motivation and a brief overview of the research methodology.

1.1. Background

The research background describes the basic terminologies and models of the research. It also describes the workflow of reaching to the problem statement.

1.1.1. Agile Methodologies

In last few decades, agile software development methodology has become one of the most popular development method due to its fast and satisfactory delivery of the product. Agile Software Development (ASD) means gathering of software development methodologies supported on iterative development, where requirements and solutions develop through cooperation between self-managing cross-working groups [1]. For numerous incremental software development procedures agile development is an expert terminology. Among agile methodology, the most popular ones are Extreme Programing (XP), Scrum, Crystal, Dynamic System Development Method (DSDM), Lean Development and Feature-Driven Development (FDD). Although there is a Uniqueness in specific slant of every agile development methodology, however the vision and principles are the same. Each agile methodology incorporate constant response in a form of feedback and reiteration for an interrupted deliverance and refinement of a software system. Agile methodologies include constant preparation, customer collaboration, responding to change and every form of progression for the project and the software [2]. In comparison to the customary waterfall-style process agile development methodologies are lightweight and innately adaptable. The most important character of agile method is its emphasis on authorizing people to work together efficiently and speedily. [3]

1.1.2. Scrum

Amongst agile development methodology Scrum is one of the leading method that is used by almost 500 companies around the world. Scrum methodology and agile principles caters more than just software development as the scrum coalition transforms the ways of handling complex projects.

Scrum guide [4] defines scrum as a lightweight, simple to comprehend and difficult to master framework by which people get to know and address challenging adaptive problems as a result create and deliver products of highest possible values. Since 1990s it has been used as a framework to cope up with complex product development. It is not necessarily a procedure for making products however, it is a framework for employment of several processes and techniques.

Scrum enhances the product management and development practices thus provides an arrangement for improvement.[4] The framework comprises of Scrum team with their specialized roles, proceedings, instructions and artifacts. Every element has an allocated purpose, which makes all the components equally essential for the usage and success of scrum framework. The events, roles and artifacts of scrum framework are bind together by the set of rules that governs the association and interaction among them [4].

1.1.3. Non-functional Requirements

Non-functional Requirements (NFRs) are the features not related to the functionality of the system. It is defined as the non-behavioral features of a system which captures the properties and limitations for the improved system`s operation.

IEEE defines NFRs as "a software requirement that describes not what the software will do, but how the software will do it, for example, software performance *requirements, software external interface requirements, design constraints, and software quality attributes" [7]*

NFRs are requisite in requirement engineering and systematic engineering since it specifies the criteria for judging the operation of a system instead of specific comportment [6]. This makes non-functional requirement contrasted with functional requirements as functional requirements only defines the specific behavior of an overall system . In short the difference between non-functional requirements and Functional requirements is, functional requirements describes what a system should do and non-functional requirements describes about how a system should be. Functional requirements usually caters an individual action of a system more so a mathematical function whereas non-functional requirements caters an inclusive property of a system as a whole or it can be overall property of a specific aspect of a system rather than a specific function. This makes non-functional requirement important as the overall properties of a system decides the success and failure of the project [6].

NFRs in software not only explains that what should software do but it also explain that how the software will do the designated goals. For instance the performance requirements, the constraints for software designing, the external interface requirements for the software and the quality attributes of the software. The functional and non-functional approaches are equally important while software development and should be handled carefully. NFRs can also be defined as extra-functional requirements like accessibility, adaptability, capability, testability, portability etc. [7]. NFRs not only represents these qualities but also signifies the constraints for system operation. Therefore, the software system will have a greater chance of success if NFRs operated carefully in the beginning of software development.

1.1.4. NFRs in Scrum

For functional requirements, agile software development methodologies are very successful however; the researchers' struggles with non-functional requirement as agile methodology do not provide clear support for NFRs management [5]. Now days the adoption of scrum and extreme programing (XP) of agile methodology is rising and there are fair chances that its usage will expands in next decade. Scrum provides an incremental delivery of difficult software solution in shorter timeframe. It also provides an improved quality and favors collaboration of the customer while ignoring documentation and tough process such as waterfall model. In agile organization, the lack of identification and modeling of non-functional requirements and its linkage with functional requirements are few of the main problems. [26]

Researchers in [8] agreed that the non-functional requirements were ill defined traditionally. During early requirement exploration phases of agile software development the NFRs were not effectively recognized, linked and modeled.

To improve the productivity of development team and speedy TTM (time to market) in software functional requirement delivery, agile software development software are used. According to the researchers in [34] the agile methodology is suffering due to not supporting NFRs as one of the primary artifact in planning and requirement analysis stages since traditionally NFRs were treated as an afterthought process. This inadequacy pertaining to NFRs is emphasized specifically while the project schedule is being planned and visualized. [27]

NFRs are often neglected due to the enormous pressure of software deploying in short time. The main focus of the software developers and the customers are on the functional approach thus neglecting the NFRs in software development process [9]. Even though NFRs plays vital role in the success and failure of the system. They not only introduce quality characteristics but also presents the limitations for operating a system usefully. NFRs are not handled carefully, thus the informal and confused handling cause major problems. [7]

1.2. GAP Analysis

The literature highlights the issue of neglect of non-functional requirements in scrum; still it lacks the in-depth analysis of the issue. As literature does not provide any knowledge regarding how non-functional requirements are currently being handled in the industry it is very difficult to analyze and identify the real issues and causes that scrum teams come across while dealing with NFRs. .

1.3. Aim and Objective

The aim of this research is to provide a set of guidelines or framework that will guide through the process of handling NFR in scrum. Also, to minimize the issues scrum teams come across in dealing with NFRs in RE phases, so that the project can be built under all possible quality constraints and eliminates the risk of failure. Research objective is to conduct a survey in order to identify the current techniques that are being used by scrum teams also to recognize the issues they confer while handling NFR. Secondly to analyze the collected data and to provide solutions in the form of set of guidelines or framework that will help dealing with the identified issues. Thirdly, to conduct a case study in a local scrum organization in order to validate the given solution.

1.4. Motivation

Due to lack of the knowledge regarding how scrum's teams manage NFRs in practice, it has become difficult to identify the challenges behind this issue. Mishandling and neglect of NFRs in requirement phases are the majorly mentioned issues in literature that leads to number of high priority problems like project failure, over budgeted and over time projects, Lots of change requests at the end of the project (some of which cannot be met at that time). Thus, it is essential to identify the real issues and problems that cause the negligence in practice. In order to provide useful techniques or methods to manage NFRs .

1.5. Research Question

RQ1: How scrum teams manage NFRs in practice?

RQ2: What are the issues they come across while dealing with NFRs during phases of requirement engineering?

1.6. Research Methodology

Research methodology started with a detailed study of literature that helps in identification of the research problem. Based on the identified problem survey in the form of interviews was planned for data collection. Interview data was than analyzed using qualitative data analysis methods and identified issues were resolved by proposing a set of guidelines. For the Validation of proposed solution, a focus group meeting was conducted and solution was improvised accordingly. Fig 1.1 below shows the steps undertaken for conducting the research.

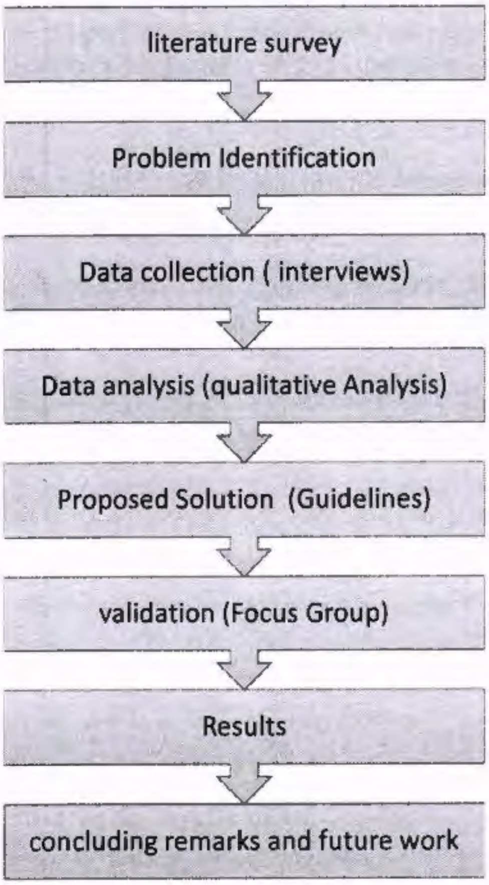

literature survey
Problem Identification
Data collection (interviews)
Data analysis (qualitative Analysis)
Proposed Solution (Guidelines)
validation (Focus Group)
Results
concluding remarks and future work

Figure 1.1: Research methodology flow chart

1.7. Thesis Outline

Thesis comprised of the chapters shown in table 1.1. description of the chapters is given below.

Chapter	Description
Chapter 2	Literature review
Chapter 3	Research Methodology
Chapter 4	Analysis and Results
Chapter 5	Proposed Solution
Chapter 6	Validation and Improvement
Chapter 7	Summary and Conclusion

Table 1.1 thesis outline

- Chapter#2: this chapter comprises of the related work in literature about management of non-functional requirements in scrum environment. It highlights the issues mentioned in literature that helps in creating the research questions for the thesis.
- Chapter#3: details of the research methodology used in conducting this research is described in this chapter. All the methods used starting form literature review, data collection and analysis up to solution validation are explained in this chapter.
- Chapter#4: This chapter comprises steps of data analysis. How data analysis is performed in order to get relevant results referring to research questions is described.
- Chapter#5: this chapter explains the solution design. It comprises of the guidelines and checklist proposed to overcome the issues also, it briefly compiles the comparison of proposed solution with other possible approaches

- Chapter#6: explains the validation process carried out in the form of focus group meeting. It comprises of the FC Findings and improvements made in the solution according to the experts opinions

- Chapter#7: this chapter presents the summary of the whole thesis and concluding remarks.

Chapter 2
Literature Review

2. Literature Review

This chapter gives an overview of the literature studied for conducting the research; it highlights the identified issues that combines to create the research problem.

Agile Scrum has been recognized as one of the most effective methods in delivering functional requirements for the system. At the same time, it is criticized by the researchers on neglect of non-functional requirements

2.1. NFRs in agile

Agile software engineering has been a popular methodology to develop software rapidly and efficiently. However, this methodology often considers Functional Requirements (FRs) due to the nature of agile software development and strongly neglects Non-Functional Requirements (NFRs). Neglecting NFRs has negative impacts on software products that have resulted in poor quality and higher cost to fix problems in later stages of software development. [11]

According to Weam et al. [12] Scrum and Extreme programming as the agile software are developing and gaining rapid popularity and are predicted to be more successful in next few years. The software engineers due to their heavy schedules are failed to give the best software, therefore they are adopting the method of short-cuts to eliminate the market's challenges. As a result of this ad hoc short cuts the quality is compromised. Precisely the non-functional requirements were neglected and not modeled properly. During early requirement phase the operationalization of non-functional requirements with functional requirement were was unidentified and ignored. Many researchers claimed that during software development phase's nonfunctional requirement was ignored and treated as the additional process. Scrum has faced difficulties as it wasn't accepted at first but at later stages it was proved to be an engineering excellence

According to Maiti et al [8]. Agile software model is a method to deliver and develop software products quickly and effectively. Agile methodology tends to favor Functional Requirements (FRs) while disregarding Non-Functional

Requirements (NFRs). Disregarding NFRs can have adverse effects on software that can result in lower quality and increased rate to repair software in later stages

M. Farid et al [26] also mentioned the same in their research, they claimed that for functional requirements agile software development methodologies are very successful however the researchers' struggles with non-functional requirement as agile methodology do not provide clear support for NFRs as a prominent choice in early software engineering phases. Now a days the adoption of scrum and extreme programing (XP) of agile methodology is rising and there are fair chances that its usage will expands in next decade. Scrum provides an incremental delivery of difficult software solution in shorter timeframe. It also provides an improved quality and favors collaboration of the customer while ignoring documentation and tough process such as waterfall model. In agile organization the lack of identification and modeling of non-functional requirements and its linkage with functional requirements are few of the main problems. For accomplishing the two main objectives that are inspection and adaptation scrum ed engineering fineness.

Taehoon et al. [15] expresses their views on NFRs management in scrum. They said that main purpose of the agile approach is the maximum production in lesser time which is done by minimizing the document and declining of the unnecessary practices. For achieving the usual focus is on the functionality of the software hence the non-functionality is neglected. The already existing agile approach is only satisfying the functional requirement and the change in improving the process requires a huge cost. Currently most of researches on quality enhancement focused on code quality and on reformation of the agile process. These researches lack the non-functional approach which is a huge draw back.

Frauke et al. [19] emphasizes on the need of management of NFRs in agile environment. They say Agile methods needs to include the handling of non-functional requirement as generally the approach does not handle the NFRs at all. According to their research the users ignores the resources, maintainability, portability, performance and safety. Most of the NFRs have to be handled carefully while the development is in progress as they affects the database choice, programing language and operating system.

Niel et al.[16] mentions Cao and Ramesh suggested that a high-quality interaction with stakeholder is needed in order to develop a software that satisfied all the requirements. They investigated that over sighting the non-functional requirements, requirement validation and scalable prototype are the main problems of agile project.

Researchers in [25] Waem et al. agreed that the non-functional requirements were ill-defined traditionally. During early requirement exploration phases of agile software development the NFRs were not effectively recognized, linked and modeled.

Some of the issues and causes of negligence of NFRs are also reported in literature that are explained below:

2.2. NFRs in early phases of SDLC

According to Elke et al. [22] The elicitation of Quality requirements at the earlier stage of the software development process is crucial as it affects an overall success of the project moreover they cannot be added at later stages. This will avoid the wrong architectural decision which will eventually discard the idea of investment loss. These quality requirement should be invented and addressed explicitly using templates as they are not available in user's minds.

According to Sachdeva et al [10] Non-Functional Requirements (NFRs) have been either mostly ignored or introduced late in the software development cycle, oftentimes leading to project failures. With the recent, increasing shifts towards cloud and emphasis in big data in the software industry, NFRs, such as security and performance, have become more critical than ever before

To improve the productivity of development team and speedy TTM in software functional requirement delivery, agile software development software are used. According to the researchers [27] the agile methodology is suffering due to not supporting NFRs as one of the primary product in planning and requirement analysis stages since traditionally NFRs were treated as an afterthought process.

This inadequacy pertaining to NFRs is emphasized specifically while the project schedule is being planned and visualized.

Elke et al [22] mentions that renowned researcher Cao and Ramesh after performing an experimental research on the agile engineering requirement suggested that knowing about the customer's requirement and to gain a uniform customer's approach are the few main challenges in agile development process. They also highlighted that the negligence of non-functional requirement is another big issue as ignoring NFRs at initial levels of the software development causes lot of challenges at later stages.

2.3. Role of Stakeholders

a) **Focus of stakeholders**

Pratima et al. [7] describes the issue of negligence of NFRs due to the pressure of software deploying in short time period. The main focus of the software developers and the customers are on the functional approach thus neglecting the NFRs in software development; even though NFRs plays vital role in the success and failure of the system. NFRs not only introduce quality characteristics but it also presents the limitations for operating a system usefully. The NFRs are not handled carefully thus the informal and confused handling of NFRs caused major problems.

Taehoon et al [15] also mentions that the main focus of the participant in agile process is on functionality of the system as it maximizes the efficiency at production level; nonetheless the security, accuracy and performance are few of the important aspects for the non-functional requirements because in later stages these can cause cost problems and waste efforts. Therefore an innovative approach is needed to deal with the non-functional aspects in agile process. In this regard a quality lightweight evaluation methods is proposed in order to enhance as well as improve the non-functional features in agile approach. This evaluation technique will allow the participants to improve the process qualitatively during each release.

b) Customers

Armin et al. [14] says that Non-functional aspects with customers is of vital importance. There are various problems caused by the non-functional aspects at the design level, whereas the problem became bigger at the implementation level in the case of agile methods. Clients rarely refers to the specifications as transaction time, confidentiality and ease in usage.

According to Alberto et al. [18], the problem lies in identification of non-functional requirements such as scalability and security because the customers often neglect them. This will eventually affects the final version. The team should guide the customer in order to avoid this hidden problem. Since agile methodologies lacks such specific techniques therefore it may be at high risk due to this non-functional approach

Borge et al. [20] also found customers as one of the reasons behind the said issue. They said that the customers are mostly unable to define NFRs regardless of whether they are using the Agile method hence the Agile RE is a weak approach. Most of the amendments based on customer`s feedback in Agile are GUI related which makes the customers unaware of the NFRs approach Ramesh et al. explained it as the major risk which should be handled carefully.

Frauke et al [17] states in their research that normally the non –functional requirements in agile process are not well defined; moreover customers are oblivious about the resources, portability, maintenance, safety or performance when they talk about what they need. The non-functional requirement affects the choice of database, programing language and operating system therefore the agile methods should be clearer in handling non-functional requirements.

According to Kvitha et al. [23] the problem lies in developing user friendly software is because in general the users or customers hardly have the idea about their functional requirement and completely ignores the non-functional requirements from the very beginning.

2.4. Techniques for managing NFRs

Irum et al [13] mentions that In Agile RE the negligence of NFRs is an important challenge because it only explains the system features. Another crucial challenge while agile development is lack of technique for operating NFRs. The results includes security factors and over costing which eventually effects the whole performance.

Alberto et al. [18] mention in his research that agile methodology lacks the specific technique for managing the non-functional requirements. An indirect approach is followed during the collection activity.

2.5. Testing of Non-functional requirements

For every software development methodology requirement verification is of utmost important. Most of the time the focus is on the functional requirement thus ignoring the non-functional ones for example performance, availability and usability[28]. These attributes needs to be tested before the actual implementation. According to most of the studies the main problems in the design are caused due to bad configuration. As the system development continues the cost of improvement or amendment increases. There is a need for a change in overall IT business which requires adaptation to the new techniques. The increase in business values of software development and decrease in cost will inspire the software developers for the adoption of different routes in project management. In software development technique the word Agile became a keyword. These are established on the continuous evolution of requirements, design and program development thus involves automatic testing of the current release. [28]

Niel maden et al. [16] investigated that over sighting the non-functionality, requirement validation and scalable prototype are the main problems of agile project.

A brief summary of the highlighted issues and factors mentioned in the following table 2.1

Sr. no.	Summary of issues/factors mentioned in literature	Reference
1.	Focus of stakeholders on functional requirements of the system	[13] [15][7][19][16]
2.	Lack of techniques for managing Non-functional requirements	[13][18]
3.	Clients fails to specify /mention all NFRs	[14][18]

4.	NFR elicitation from clients is difficult	[14]
5.	Customers are unaware of NFRs	[17][20][23]
6.	Lack of NFR testing	[28][16]
7.	Configuration Problems difficult to manage at the end	[28][15]
8.	Lack of NFR elicitation /identification in the beginning	[14][15][22]
9.	Lack of NFR planning	[27] [12]
10.	Lack of NFR analysis	[27][12]

Table 2.1 summary of literature

2.6. Literature Review Analysis.

Above-mentioned literature shows many articles that highlights the issue of negligence of non-functional requirements in scrum environment. Table 2.2 below shows the frequency of occurrence of issue causes consequences and solutions mentioned in literature.

However, some limitations can be observed. Studies shows that they do not cover all requirement phases or a complete software development lifecycle. In other words most of the studies are limited to one or two phases hence lack the issues NFRs have in other phases. Also, literature lacks the detailed analysis of the consequences and causes of the highlighted issues that makes it difficult to analyze the rea problem.

Secondly, it is also observed that the issues and causes mentioned lacks industrial survey that is most of them do not mention the real issues industry is facing for managing NFRs and the real causes behind the issues. Especially in the context of Pakistan where scrum industry is rapidly growing and gaining lots of importance, we see lack of studies (research) regarding this issue.

Most importantly, no study is found that gives a complete set of solution for overcoming the issues in all scrum phases or covers all RE phases in complete scrum process.

Hence through this study, we tried to present a solution to overcome the issue in management of NFRs so that we can be saved from after effects.

Paper ID	Issues of NFR negligence Addressed	Causes of negligence Addressed	Consequences of issues Addressed	Proposed Solution
[13]	✓	✓		
[14]	✓	✓		
[15]		✓	✓	✓
[16]	✓	✓		
[17]		✓	✓	
[18]		✓		
[19]		✓		
[20]		✓	✓	
[21]		✓	✓ (generalized)	
[22]	✓			
[23]		✓		
[24]				
[25]	✓			✓ (single issue addressed)
[26]	✓			✓ (single issue addressed)
[27]	✓			✓ (single issue addressed)
[28]	✓			

Table 2.2 : literature review analysis

Chapter 3
Research Methodology

3. Research Methodology

This chapter gives overview of the reasearch design, comprised of research methodolgy, phases and approach used to conduct the research. Fig 3.1 shows the process flow of conducted research.

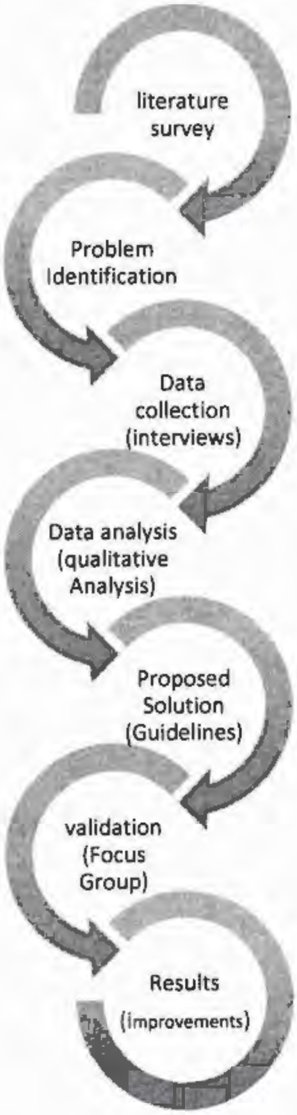

literature survey
Problem Identification
Data collection (interviews)
Data analysis (qualitative Analysis)
Proposed Solution (Guidelines)
validation (Focus Group)
Results (improvements)

Figure 3.1 : Research phases

3.1. Literature Survey

Review of literature was the first step carried out in conducting this study. Different digital libraries were explored in order to identify the literature on management of NFRs in scrum environment. These libraries includes IEEE Xplore, ResearchGate, SpringerLink, ACM digital library .

3.2. Problem Identification

Survey of literature leads us to the problem of lack of non-functional requirements handling in agile (scrum) enviornment with in RE phases. Which is discussed in detail in chapter 2.

3.3. Data Collection

Semi structured interviews [29] is one of the most popular techniques for qualitative data collection. This technique has various advantages as it ensures deeper understanding and insights of the context. Also, use of an interview guideline ensures a common structure thus making results comparable [29]. Thus, the technique was selected for collection of data in this study.

The data collection process consisted of three phases shown in figure (3.2) described below:

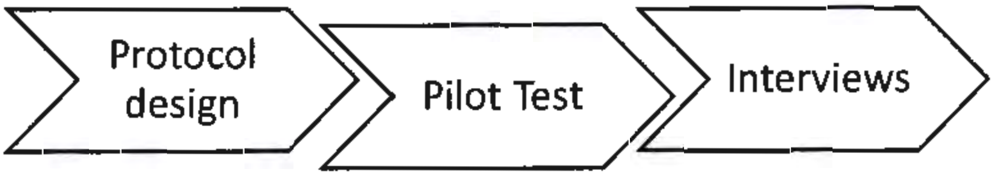


Figure 3.2: Data Callection Process

deeper insight about how NFRs are managed w.r.t requirement phases (elicitation , specification etc) [25] and to reveal causes and consequences of mishandling of NFRs in scrum organization. Total duration of interviews was between 45 to 50 minutes. In order to make sure that no information is lost *taping* was used to record the interviews. Excel was used as the tool to transcribed data. Details are given in chapter 4.

3.4. Data Analysis

For analyzing the qualitative data gathered from survey interviews thematic network analysis technique[31] and root cause analysis [33] was used.

- **Thematic network analysis**

A thematic network analysis [31] approach is best suited for the in-depth studying and understanding of tentative interview data. For investigation and identification of the hidden meanings and implicit subjects in an interview, thematic analysis is specifically suitable. Firstly the text is coded with different concept and later those concepts are grouped into categories. By doing this common themes and patterns are explored. Themes can always be classified into different levels from elementary to comprehensive, like basic, organized and global themes. The most elementary theme is the basic. Textual data is the analyzed by the lowest level i.e. basic theme. Basic theme needs to be studied in the context of other themes as they do not provide significant meaning to the data as it only represent the statement of belief which is affixed in the text moreover basic themes are mostly grouped with other themes which will help in better understanding of the basic themes. The organizing theme will provide the meaning of the related basic themes. It is middle level theme that summarized the basic themes in relation with other themes at a broader level. Lastly the organizing themes dealt with the global themes. Global themes helps to understand the text as a whole, it is a significant theme that incorporates the important metaphors of the data provided.

Thematic network analysis approach with the four stage qualitative data analysis process by Rene and Taylor-Powell (2003) is applied to analyze the interviews in this research.

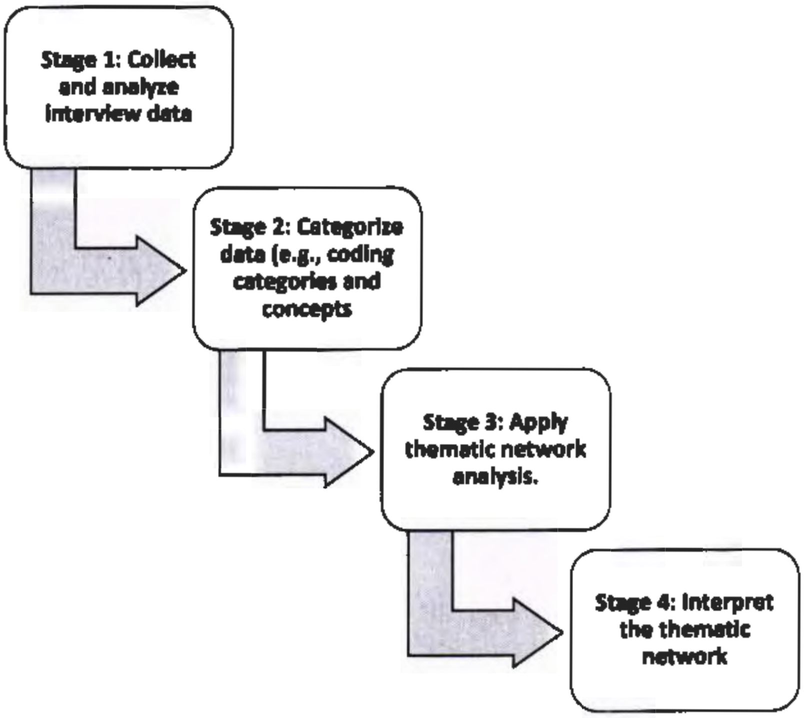

Stage 1: Collect and analyze interview data
Stage 2: Categorize data (e.g., coding categories and concepts
Stage 3: Apply thematic network analysis.
Stage 4: Interpret the thematic network

Figure 3.3. Thematic analysis process flow

Chapter 4 has details of how thematic network analysis is used to identify

a) Process of managing NFRs
b) Issues in management of NFRs

➢ **Root Cause Analysis**

After identification of issues, Root cause analysis was performed next. (RCA) [33] is a technique used to get to the underlying causes of a problem thus facilitating most effective solution. There are three fundamental things identified in order to perform RCA:

a) The problem
b) Reason for the occurrence

c) Preventive measures for the future

3.5. Proposed Solution

In order to overcome the RCs identified in the previous step and to eliminate the major issues a set of guidelines was designed that covers the whole SDLC. Proposed solution provides a checklist that helps PO to ensure management of NFRs in complete scrum Process, thus dealing with all major issues.

3.6. Validation

Once solution was designed, it was validated via a focus group of experts. Focus Group [35] is a research method that along with other uses can be used for evaluation of potential solutions, based on practitioner or user feedback. , the focus group method is a cost-effective and quick empirical research approach for obtaining qualitative insights and feedback from practitioner [33] Details are provided in chapter 6

3.7. Results (Improvements)

Experts of the focus group session approved the solution with minor changes / additions in the proposed set of guidelines. Major addition (with consensus of all participants) was in the supporting checklist of quality attributes. Details are provided in chapter 6.

Chapter 4
Data Analysis and Results

4. Analysis and Results

After data collection through interview, next step was to analyze the data and get results. Figure 4.1 shows the steps undertaken for analysis of data. Analysis of data was designed in order to get the results according to the research questions.

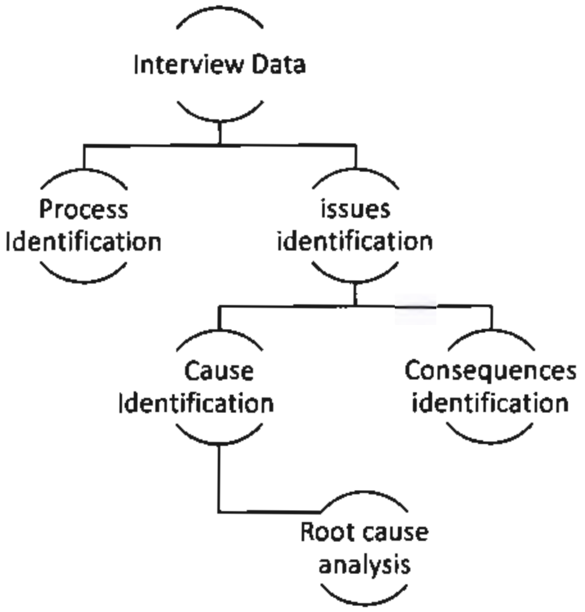


Figure 4.1: Data analysis steps

4.1. Process identification

Referring to RQ1 thematic analysis [31] applied on the interview data to find out the process organizations are following for NFR management. First of all coding technique was applied on the data in order to group it into manageable chunks called concept categories. A concept category represents the underlying concepts or knowledge embedded in the interview text. In total, we divided the data into stakeholder roles, activities and artifacts for every requirement phase. Concepts were than analyzed in order to create pattern out of each category and presented them in form of basic themes. "A basic theme [31] refers to the lowest-order premises evident in the coded text". Basic themes were than further analyzed to group into organizing themes. Basic themes that share same meaning issues arguments or assumption are grouped under one cluster i.e. organizing theme. For instance we grouped basic themes into process, techniques, artifacts and roles that affect NFR management. Which were then further reduced to three Global themes (highest level themes). Stakeholder-based, activity –Based and artifacts Based. Complete Thematic network analysis is shown in appendix [C]and appendix [D] issues were identified than from the text referring to RQ2.

4.2. Issues- Identification

❖ **Thematic analysis**

Global themes of network analysis of process identification were used as the concept categories for the issues (referring to RQ2). Again, thematic network analysis is performed on the data in order to find out the major issues. Appendix [D] shows the code categorization and basic themes division for issues identification. Table 4.4 below shows the organizing and global themes . Results (major issues) obtained from this thematic analysis network are compiled in table 4.5

issues. Basic themes	**Organizing themes**	**Global themes.**
Lack of initial NFR identification by clients. Identification of NFRs after deliverables. Focus of stakeholders on functional requirements. Initially. Lack of NFR related questions in requirement elicitation questionnaires	NFRs are overlooked during requirement elicitation process.	**Process based**

Lack of explicit methods for NFR elicitation Lack of awareness about NFRs.		
The change request docs miss NFRs addition. User stories for NFRs are difficult to create No explicit user stories are created for NFRs No explicit method for specification of NFRs	NFR specification is considered difficult	**Technique-based**
NFRs are not discussed in every sprint. Lack of NFR discussion by PO throughout the project. Reliance on developer's experience for NFR development throughout the project.	NFR are not discussed in every sprint	**Process -based**
Lack of NFR prioritization Product backlogs lacks NFR related items. Cost estimation and resource allocation for NFRs remain indeterminate.	NFRs are not part of Product Backlog.	**Artifact-based**

Test cases for only explicitly mentioned NFRs. Explicit test cases are not written for Nfrs only. Lack of methods of NFRs testing. Not following proper methods for testing NFRs Most of the NFRs remain untested. Only standard based NFR testing Reliance on Standard for NFR testing	Inappropriate methods for NFRs testing	**Technique –based**
Reliance on organizational quality standard for NFRS development. Quality standard does not cover all NFRS. One quality standard for all types of project Org. quality standard lacks inclusion of all NFRS. Organization's own quality standard. No proper format for creating quality standard	Inadequate quality standard for NFRs.	**Artifact –based**

Table 4.4 Issues Basic, organizing and global themes

Sr.no.	Major Issues
1	NFRs are overlooked during requirement elicitation process.
2	NFR specification is considered difficult
3	NFR are not discussed in every sprint
4	NFRs are not part of Product Backlog.
5	Inappropriate methods for NFR testing
6	Inadequate quality standard for NFRs.

Table 4.5: major issues

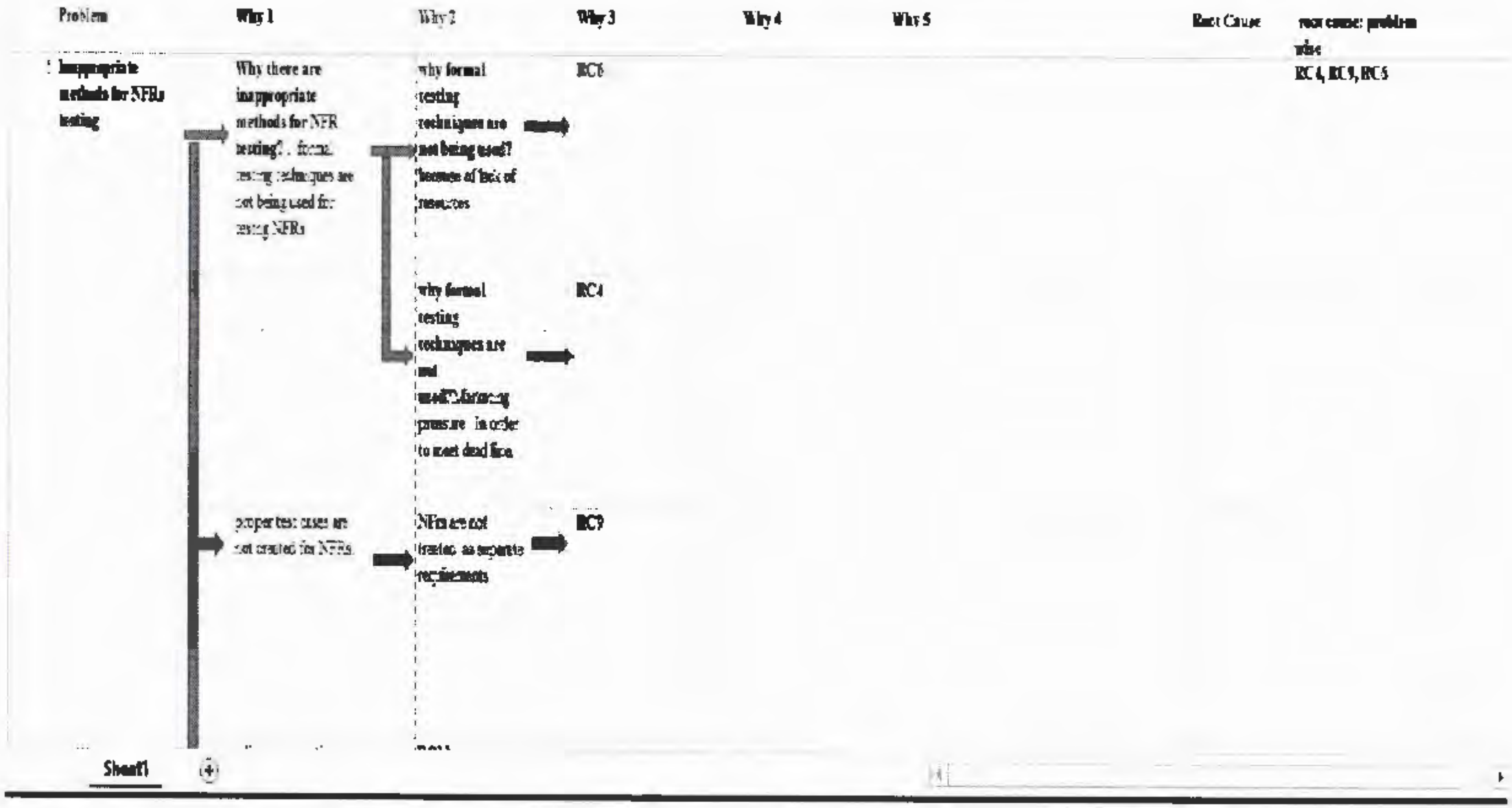
4.3. Causes of the major issues:

After identification of the major issues next step was to identify the causes of the issues. In order to eliminate the causes and built a solution that results in solving the major issues root cause of every issue was required. For which root cause analysis with the help of 5-why technique was performed.

- **Root cause analysis :5- why technique**

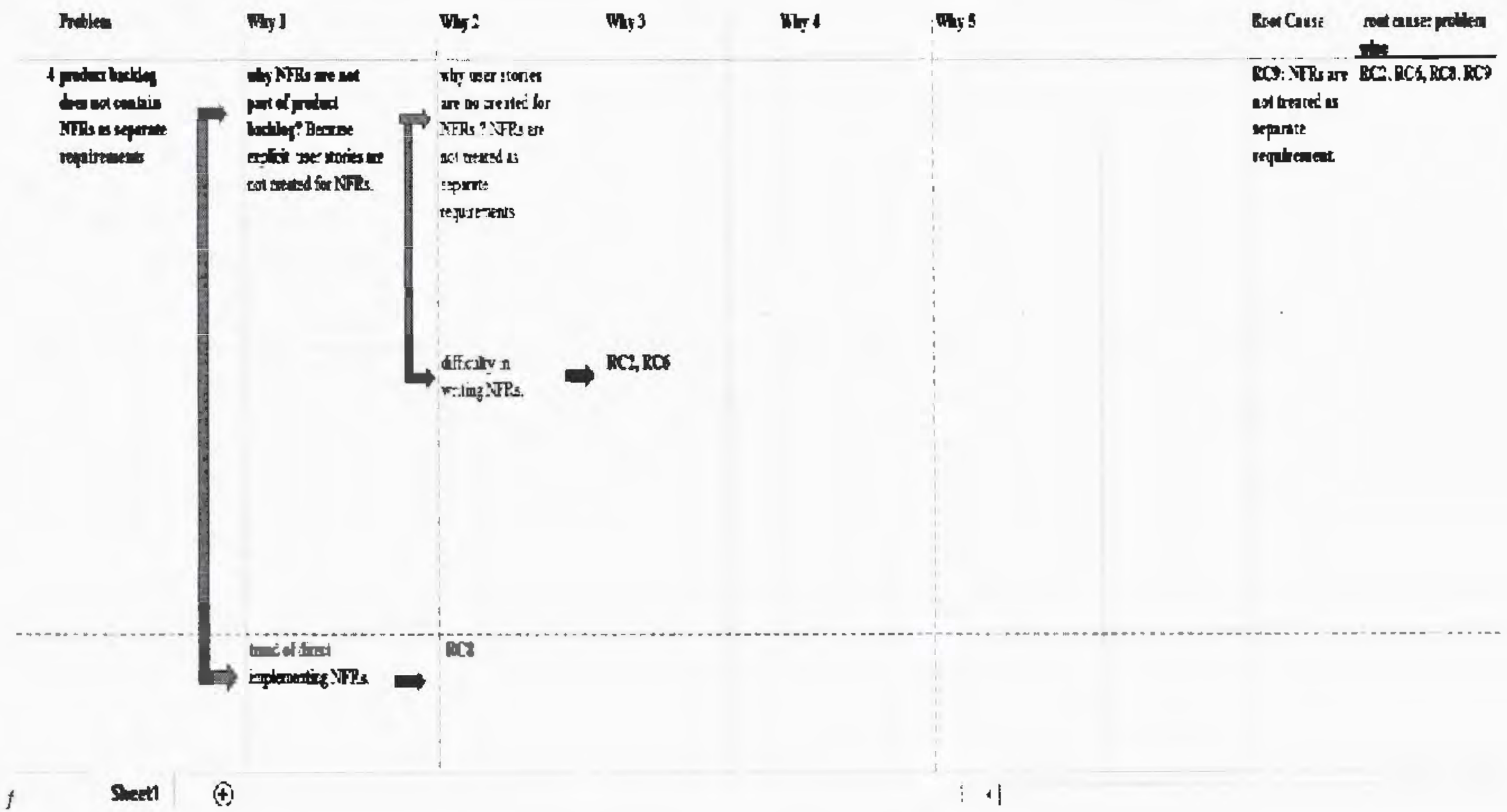
In order to find the root causes for all major issues we used 5 why technique [33]. In which every major issue was answered cause wise up to the depth of 5th level. For instance issue no. 1: NFRs are overlooked during requirement elicitation process is answered for every possible cause. Than every cause is again examined for the deeper reason of occurrence until the root cause is identified (that cannot be further divided). Figure 4.2 to 4.7 below shows the excel work for the RCA of all the major issues. Resulting in identification of root causes of all the issues.

After root cause identification raw graph tool [38] is used to express the relationship between major issues and their root causes. Figure 4.9 shows the relationship diagram of the issues with their root causes. Different colors are used for every issue and width of the colors represent the no. of root causes associated with the issue.



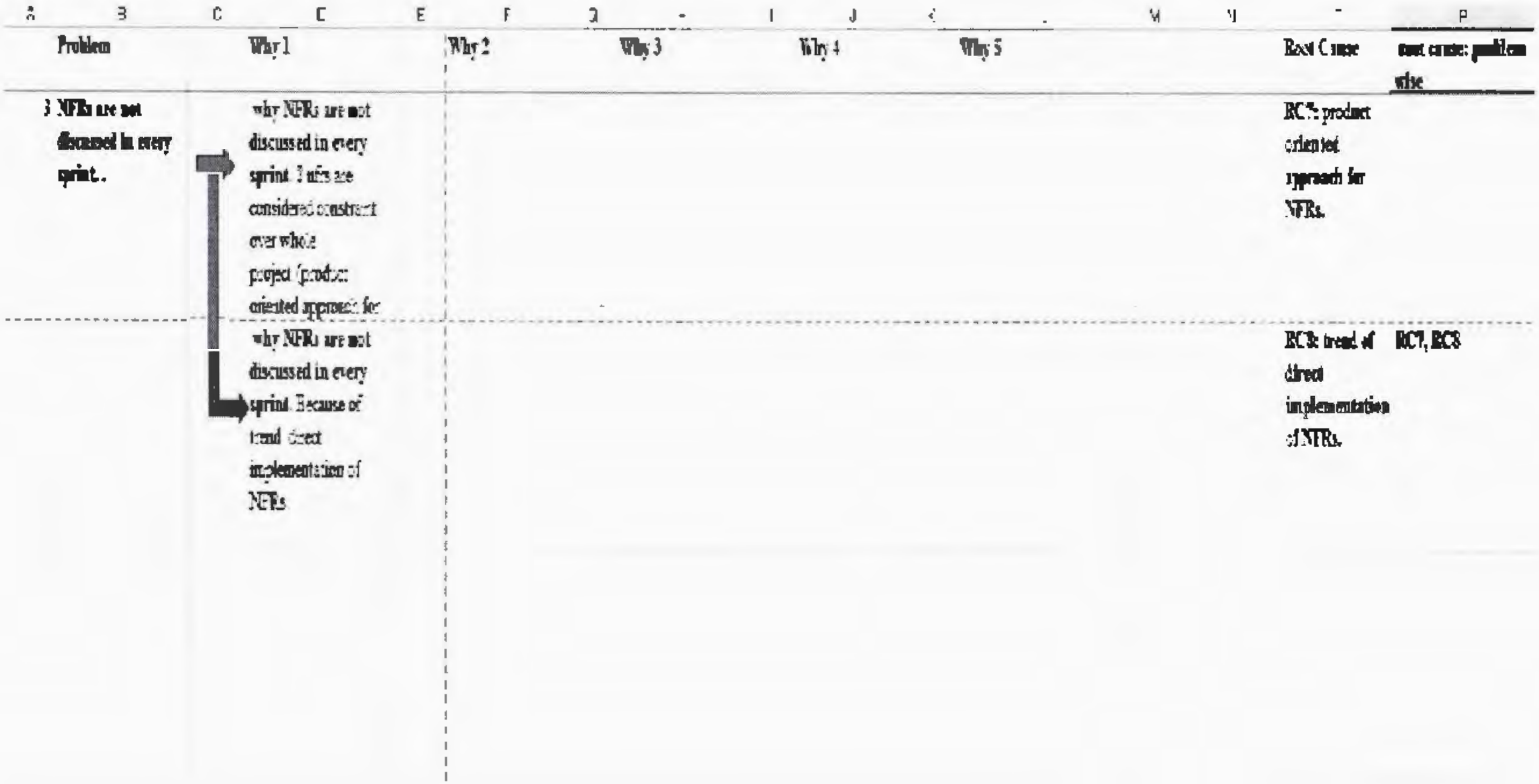
Problem	Why 1	Why 2	Why 3	Why 4	Why 5	Root Cause	root cause: problem wise
Inappropriate methods for NFRs testing	Why there are inappropriate methods for NFR testing? . formal testing techniques are not being used for testing NFRs	why formal testing techniques are not being used? because of lack of resources	RC6				RC 4, RC 9, RC 6
		why formal testing techniques are not used?because pressure in order to meet dead line	RC4				
	proper test cases are not created for NFRs.	NFRs are not treated as separate requirements	RC9				

Figure 4.2: Screen shot 5 why analysis (1)



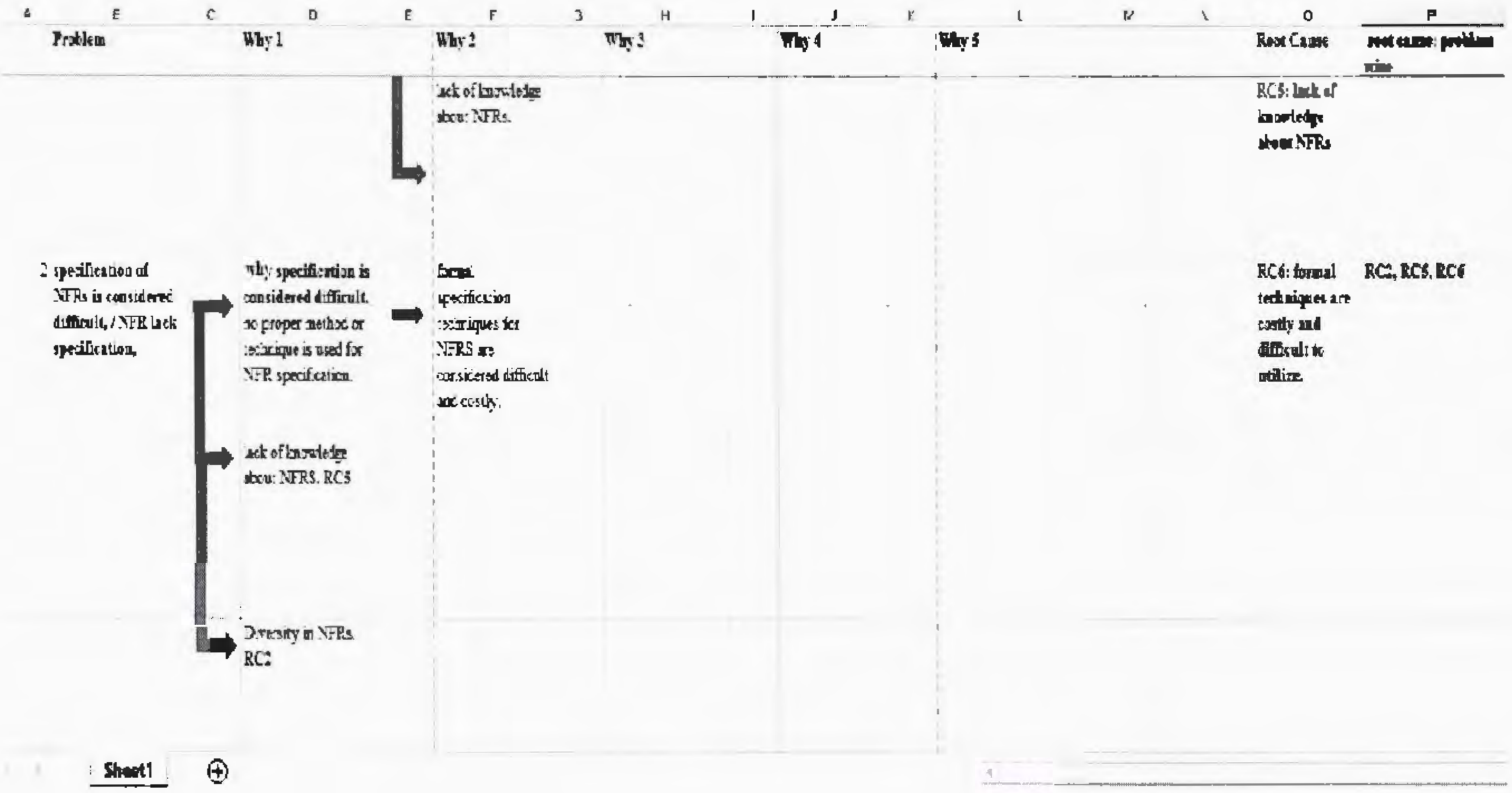
Problem	Why 1	Why 2	Why 3	Why 4	Why 5	Root Cause	root cause: problem wise
4 product backlog does not contain NFRs as separate requirements	why NFRs are not part of product backlog? Because explicit user stories are not created for NFRs.	why user stories are no created for NFRs ? NFRs are not treated as separate requirements				RC9: NFRs are not treated as separate requirement.	RC2, RC6, RC8, RC9
		difficulty in writing NFRs.	RC2, RC6				
	time of direct implementing NFRs.	RC8					

Figure 4.3. Screen shat 5 why analysis (2)



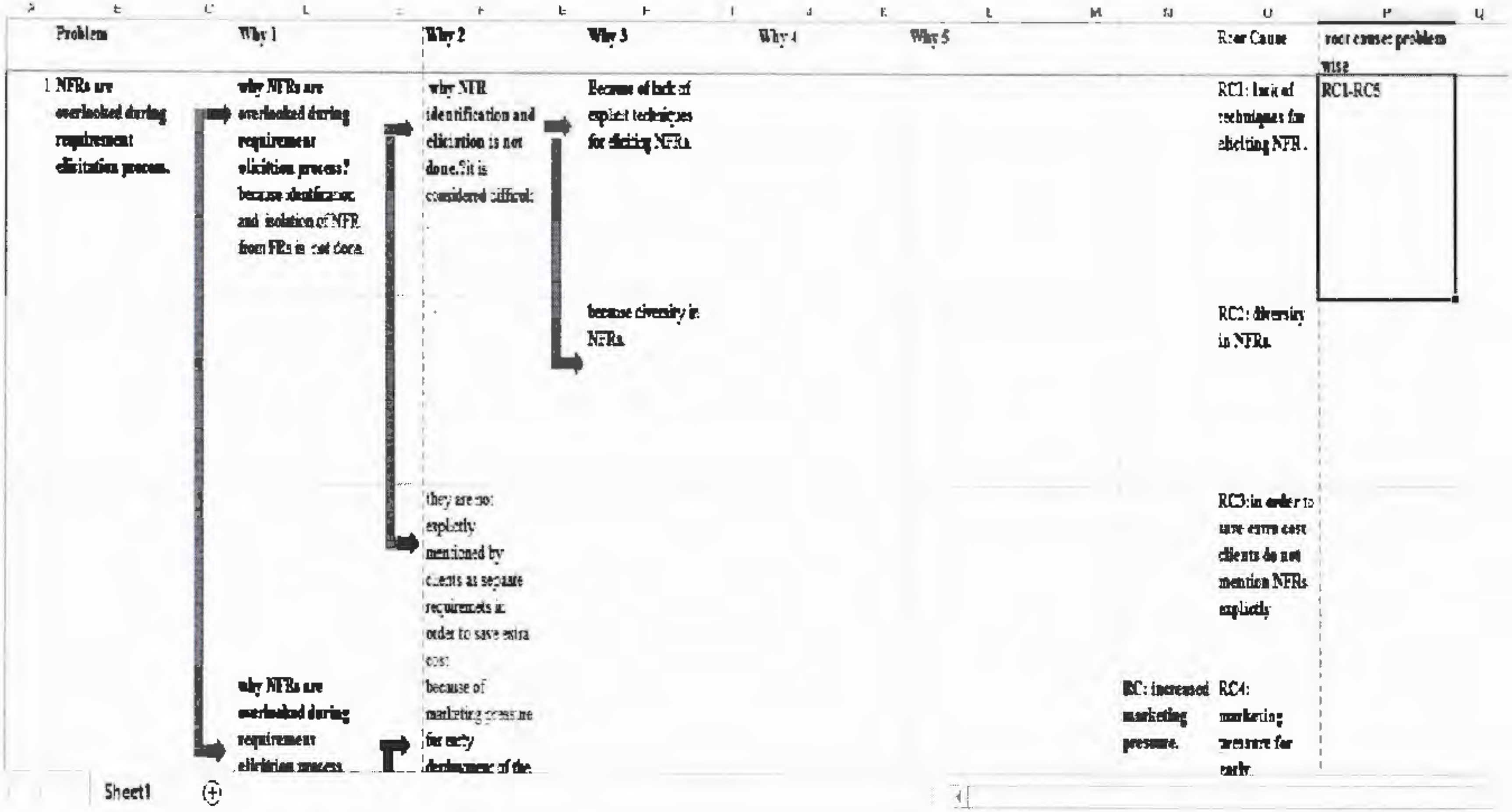
Problem	Why 1	Why 2	Why 3	Why 4	Why 5	Root Cause	root cause: problem wise
3 NFRs are not discussed in every sprint..	why NFRs are not discussed in every sprint. 3 nfrs are considered constraint over whole project (product oriented approach for					RC7: product oriented approach for NFRs.	
	why NFRs are not discussed in every sprint. Because of trend direct implementation of NFRs					RC8: trend of direct implementation of NFRs.	RC7, RC8

Figure 4.4 : Screen shot 5 why analysis (3)



Problem	Why 1	Why 2	Why 3	Why 4	Why 5	Root Cause	root cause: problem wise
		lack of knowledge about NFRs.				RC5: lack of knowledge about NFRs	
2 specification of NFRs is considered difficult, / NFR lack specification,	why specification is considered difficult. no proper method or technique is used for NFR specification.	formal specification techniques for NFRS are considered difficult and costly.				RC6: formal techniques are costly and difficult to utilize.	RC2, RC5, RC6
	lack of knowledge about NFRS. RC5						
	Diversity in NFRs. RC2						

Figure 4.5: screen shot 5 why analysis(4)



Problem	Why 1	Why 2	Why 3	Why 4	Why 5		Root Cause	root cause problem wise
1 NFRs are overlooked during requirement elicitation process.	why NFRs are overlooked during requirement elicition process? because identification and isolation of NFR from FRs is not done.	why NFR identification and elicitation is not done.?it is considered difficult	Because of lack of explicit techniques for eliciting NFRs.				RC1: lack of techniques for eliciting NFR .	RC1-RC5
			because diversity in NFRs.				RC2: diversity in NFRs.	
		they are not explicitly mentioned by clients as separate requirements in order to save extra cost.					RC3:in order to save extra cost clients do not mention NFRs explictly	
	why NFRs are overlooked during requirement elicitation process.	because of marketing pressure for early development of the				RC: increased marketing pressure.	RC4: marketing pressure for early.	

Figure 4.6: Screenshot 5 why analysis (5)

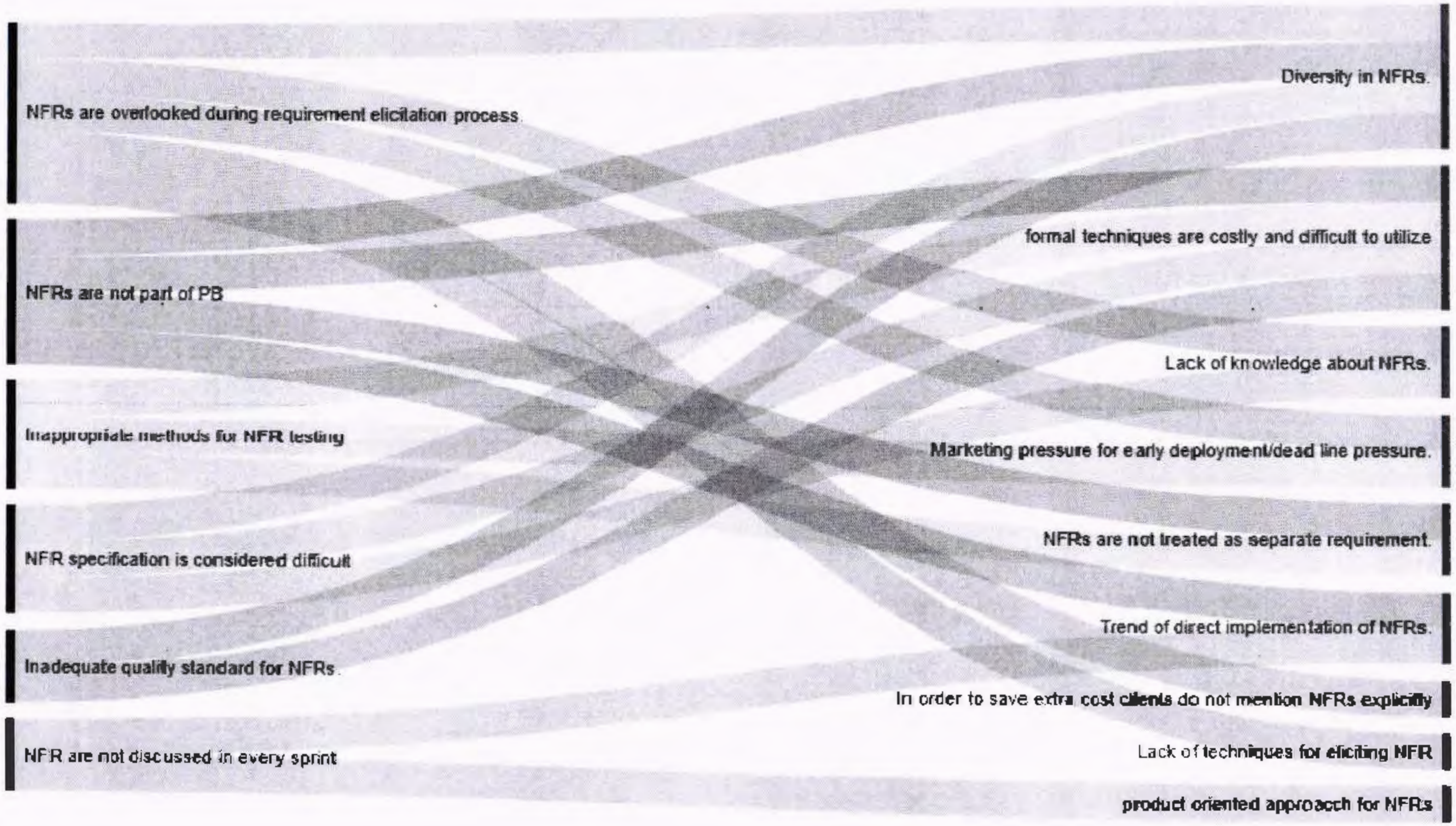

NFRs are overlooked during requirement elicitation process.
NFRs are not part of PB
Inappropriate methods for NFR testing
NFR specification is considered difficult
Inadequate quality standard for NFRs.
NFR are not discussed in every sprint
Diversity in NFRs.
formal techniques are costly and difficult to utilize
Lack of knowledge about NFRs.
Marketing pressure for early deployment/dead line pressure.
NFRs are not treated as separate requirement.
Trend of direct implementation of NFRs.
In order to save extra cost clients do not mention NFRs explicitly
Lack of techniques for eliciting NFR
product oriented approacch for NFRs

Figure 4.7: issue-cause relationship diagram

4.4. Consequences of the issues

In a software development life cycle, all the phases of software development are interconnected with one another, so there is a huge chance that problem i one phase create issues in other phases. As we can observe links in our identified issues. It is the case with consequences. For identified issues consequences are repetitive or in other words we can say that many issues share common consequences shown in table 4.11 below.

Sr.no.	Major Issues	Consequences
1	NFRs are overlooked during requirement elicitation process.	• Arbitrary design choices regarding NFRs. • Missing of requirements • Rework • Extra time effort ad cost.
2	NFR specification is considered difficult	• Things that are not defined are easily missed • Testing of requirements whose test cases are not written can easily be neglected.
3	NFR are not discussed in every sprint	• Arbitrary design choices. • Missed from product backlog and sprint backlog.
4	NFRs are not part of PB.	• Missed requirement traceability
5	Inappropriate methods for NFR testing	• Lots of change requests regarding NFRs. • Lots of Rework • Over budgeted projects.

		• Few changes cannot be met at the end of the project hence can result in project failure
6	Inadequate quality standard for NFRs.	• Missing of some major requirements • Late identification of some early need to be implemented requirements. • Incomplete testing of project if testing of NFR is about checking in accordance with quality standard.

Table 4.11: consequences of the major issues

Chapter 5
Proposed Solution

5. Proposed Solution

This chapter explains the proposed solution in detail. Solution consists of three steps as shown in figure 5.1

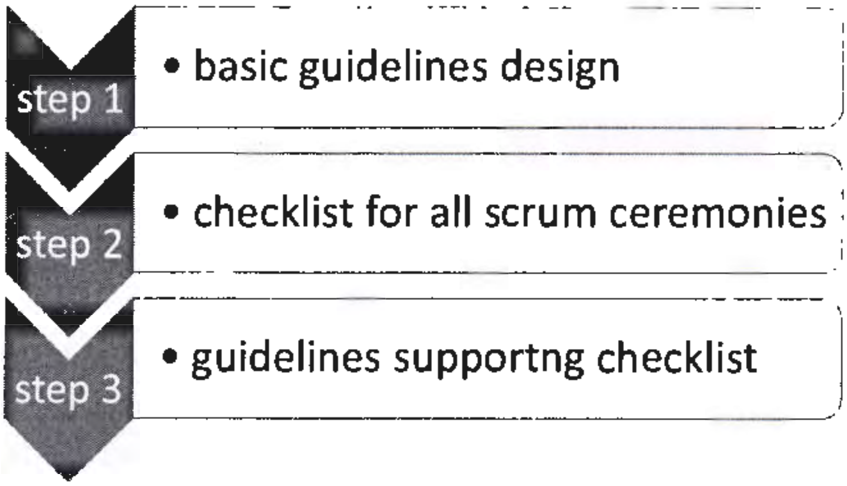

step 1
• basic guidelines design
step 2
• checklist for all scrum ceremonies
step 3
• guidelines supportng checklist

Figure 5.1 solution construction

5.1. Step 1: Basic guideline design

In order to solve the major issues mentioned in table 4.5 we designed a set of basic guidelines that covers all major issues and helps in improving the process.

➢ **Early Identification and quantification OF NFR**

1. Elicitation of NFRs at initial levels with functional requirements.
   1.1. Directly from customers by adding questions about NFRs in requirement gathering **questionnaire**
   1.2. Using **prototypes.**
2. Identification of NFRs from elicited Functional requirement taking help from ISO quality standard.
3. Identifying the sub quality attributes with the help of quality standard of the elicited quality attributes.
4. The NFRs collected from FRs than are to be discussed by clients and priority can be set in consensus with them.

➢ Product backlog management with NFRs.

1. Elicited NFRs now should be added as separate items in product backlog with labeling as constraint 1, constraint 2, and constraint 3...

   [These separately mentioned NFRs have two conditions.

   - One NFR can be constraint over multiple user stories (functional requirement)
   - One user story (US) can have multiple constraints.]

2. For the purpose of NFR management in product backlog NFRs will be added separately in

   2.1.Acceptance criteria of every user story

   2.2.Definition of done as a separate requirement.

For example:

Acceptance criteria for US1:

Feature 1, feature 2

Constraint 1, constraint 3

So a story is not accepted unless constraints are not achieved.

Similarly

Definition of done for constraint 1:

US 1, US3, US5 should be done.

3. A constraint that belongs to a story form next iteration will stay in definition of done but completing its part of 1 user story the acceptance criteria will ensure the completion of 1 user story

➢ Iterative and Continuous testing of NFR

1. The concept of iterative testing is to be introduced, that is to test NFRs in every iteration whenever and at any possible rate.
2. Testing of every sub quality attribute will make iterative and continuous testing possible.

3. Using ISO standard 9126 as a check list for testing will make process easier and authentic.
4. Test cases must be written for every constraint and should be dealt in every possible iteration.
5. Using Prototypes as suggested earlier can also help in verification of NFRs.

➢ **Role of Product Owner**

1. *Need to introduce transparency in requirements.*
2. *ISO standard has been followed all over the SDLC from identification to validation?*
3. *NFRs are considered and discussed in every sprint?*

5.2. Step 2: Checklist for all scrum ceremonies

In order to execute the solution in industry the basic guide is than converted in the form of a checklist for all scrum ceremonies i.e. product backlog creation, sprint planning, daily scrum etc. [4]. Starting from requirement identification and ending on to sprint retrospective set of checklist is provided for all scrum events. Appendix [A] provides the complete description about the scrum events for which guidelines are created

5.3. Step3: Supporting guidelines for checklists

In order to get the tasks done mentioned in checklists supportive guidelines are provided for checklist so that teams will not have any problem for achieving the goals of checklists.

Following is the finalized version of Solution (checklist along with their supportive guidelines)

❖ **Scrum Phases**

1. Pre- Requisite to Sprint planning

a. Requirement Identification

b. Product Backlog creation

2. Sprint Planning
3. Daily scrum
4. Sprint review
5. PB Grooming
6. Sprint retrospective

1. Pre –Requisite to Sprint planning

a. Requirement Identification

2. Product Backlog creation

Requirement identification checklist		
1	NFRs are elicited from customers during initial requirement elicitation?	❐
2	Constraints over all functional requirements are identified?	❐

Guidelines for requirement identification

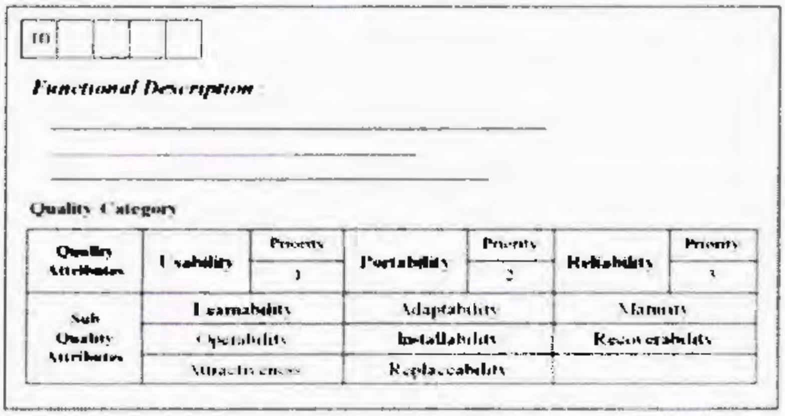
1. PO should include NFR related questions early on while eliciting requirements. A checklist provided (see table 5.1(I)), that is created according to ISO standard 9126-[43] will help PO in identification of NFRs.
2. Take help of prototypes in order to make clients understand NFRs easily.
3. Every feature should be analyzed quality wise in order to find out constraints over them.

Product backlog creation checklist		
1.	NFRs are separately mentioned in user stories	❐
2	NFRs are labeled and prioritized	❐
3.	NFRs are categorized according to given standard (checklist)	❐

Guidelines for creating product backlog

1. While writing user stories constraints over functional requirements should be mentioned along

An exemplary card given below:



ID

Functional Description

Quality Category

Quality Attributes	Usability	Priority	Portability	Priority	Reliability	Priority
		1		2		3
Sub Quality Attributes	Learnability		Adaptability		Maturity	
	Operability		Installability		Recoverability	
	Attractiveness		Replaceability			

Explanation of the card:

i. ID is the number assigned to user story, for example US 1, US2 etc.

ii. Functional description is the feature or functional requirement.

iii. The quality category are the NFRs associated with the above mentioned features

iv. Sub quality attributes are decided with help of ISO standard 9126-1

v. NFRs are labeled as constraint C1, C2, C3 and so on.

2. Sprint Planning

Sprint planning checklist		
	Sprint goal	□
1	Sprint goal define quality measures of the sprint?	□
	Test cases/DoD	□

2	Test cases are written for NFRs?	□
3	Acceptance criteria of each user story has been made w.r.t its constraints?	□
4	NFRs are added into the done criteria?	□
5	Test criteria for NFRs is planned Iteratively?	□
	Value estimation	□
6	Cost estimation of a requirement includes its NFRs?	□
7	Schedule estimation for a requirement is done including NFRs?	□

Guidelines for Sprint Backlog

1. PO should make sure that identified NFRs are transparent and equally understood by everyone.(during requirement discussion)
2. While making sprint goal team should check the following points
   - Sprint goal is not compromising Quality of the system
   - Quality of the requirements of the current sprint are to be included in the sprint goal and not left for future sprints or end of the project

3. Acceptance criteria for a user story should look like:

Acceptance criteria of US1	
Feature 1	Description
Feature 2	Description
Constraint 1	Description
Constraint 2	Description

4. Definition of done of a user story

Definition of done for US1	Done
Feature 1 tested/done	□
Feature 2 tested /done	□

Other factors …	□
Constraint 1 tested /done	□
Constraint 2 tested/ done	□

5. Quality testing should be done alongside with feature testing.(During sprint)

3. Daily scrum

Daily scrum checklist		
1	**What did you do yesterday?**	□
2	Done quality measures are mentioned?	□
3	**What will you do today?**	□
4	Today's work mentions quality attributes?	□
5	**Are there any impediments in your way?**	□
6	Quality related impediments are mentioned?	□

Guidelines for daily scrum:

Questions to be asked in daily standup.

- Did you achieve all quality factors (mentioned) related to yesterday's work?
- What quality factors you are currently working on??
- What are the factors that are refraining from achieving quality factors?

4. Sprint review

Sprint review checklist		
1	Done NFRs are mentioned explicitly in the meeting?	□
2	NFRs that could not get completed are mentioned?	□
3	Factors that help in achieving sprint quality are explicitly mentioned?	□
4	Quality attributes for upcoming sprint are discussed explicitly?	□

5	Calculation of remaining work mentions remaining NFRs?	□

Guidelines

Questions of review should add quality related questions

1. Quality of the sprint has been achieved expectedly.

2. What factors create hurdles in achieving quality on time?

5. PB Grooming

Product backlog grooming checklist		
1.	User stories with incomplete constraints are added up in backlog for next iteration?	□
2.	Done definition of incomplete NFRs remains in the PB. ?	□

Guidelines for PB grooming*:*

Above-mentioned acceptance criteria and Dod (tables) will help in identification of remaining NFRs

6. Sprint retrospective

Sprint retrospective checklist		
1.	NFRs that were well accomplished are mentioned?	□
2.	Areas where quality improvement is required are discussed and sorted?	□

Guidelines

1. Discussion about quality improvement and quality handling will guide in easy management of NFRs in later Sprints.

Table 5.1 NFR checklist

Nonfunctional Requirements				
1.	**Functionality**			□
		Suitability	□	
		Accuracy	□	
		Interoperability	□	
		Standards	□	
		Security	□	
2.	**Reliability**			□
		Fault tolerance	□	
		Maturity	□	
		Recoverability	□	
3.	**Usability**			□
		Understandability	□	
		Learnability	□	
		Operability	□	
4.	**Efficiency**			□
		Time Based efficiency	□	
		Resource based efficiency	□	
5.	**Maintainability**			□
		Analyzability	□	
		Changeability	□	
		Stability	□	
		Testability	□	
6.	**Portability**			□
		Adaptability	□	
		Instalability	□	
		Conformance	□	
		Replaceability	□	

5.4. Justification with other Approaches.

As for every problem there are multiple Solutions, similar in our case, we considered multiple Solutions for solving the aforementioned issues and selected the best according to our knowledge. A brief description of the limitations of the other possible approaches given below.

1. **Creating an NFR checklist or Matrix.**

Explanation: All non-functional requirements are specified in a form of a checklist or matrix that will help the team to ensure the identification and

validation of the requirements.

Limitations of the approach:

- This will be an informal method that does not support NFR definition/ detailed specification.
- Addition and management of NFRs in Product Backlog will remain unresolved.
- Requirement traceability (NFRs with FRs) will be missing.

Above mentioned limitations states that this technique does not support solution for all identified issues. Though identification and validation can be done up to some extent but other identified issues like specification and PB management remain unresolved.

2. **Using requirement specification tools for NFRs also.**

Explanation: Tools that help in specification of requirements are used to specify functional requirements, NFRs should be specified separately and with the same level of detail.

Limitation of the approach:

- This approach might result in excessive documentation that is not an ideal approach to work with agile documentation.

- The approach does not ensure the complete identification of NFRs, if NFRs remain unidentified this approach would not be able to figure out, and requirements will remain missing.
- The approach also does not ensure the requirement management in product backlog that is one if the identified issues.
- This might requires extra training in order to understand the tool.

Above mentioned limitations states the identified issues, hence showing that this solution is not meeting our goal of problem solving.

3. Introducing NFRs in UML diagrams.

Explanation: definition of NFRs in UML format along with FRs. Just like in NFR framework (reference no.) introduces concept of loose cases for non-functional requirements.

Limitations of the approach:

- This approach does not ensure complete identification of all NFRs
- This will increase documentation, that is contrary to agile methodologies.
- Extra training would be required in order to draw the requirements.
- This approach is limited to specification, PB management issues remain unresolved.

Chapter 6

Validation and Improvement

6. Validation of proposed Solution

This chapter explains the steps carried out for validation of proposed solution. In order to validate our proposed solution, a focus group [41] session was planned and conducted in which some of the agile experts from the industry were invited. Experts gave their opinions on the proposed solution and improvements were decided in consensus with all experts.

Below are the steps carried out for validation through Focus group.

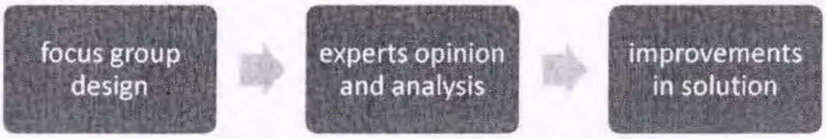


Figure 6.1: validation process

6.2. Focus Group design

- **Focus group Purpose statement**

The purpose of the focus group process is to conduct a session

A. to get opinion of agile experts on proposed set of guidelines& checklist in order to overcome the issues in negligence of NFRS with in scrum environment

B. Participants will comment on strengths and weaknesses of proposed solution and provide suggestions for specific improvements.

C. Improvements will be finalized after discussion and in consensus with all experts

- ***Demographic data:***

  Anonymous demographic data collection from focus group participants.

- ***Data collection:***

  Audio/video-taped focus session with participants agreement, and transcribed verbatim for analysis. Strictly preserved confidentiality, except where disclosure is mandated by a court of law.

- **Guiding questions**

  Guiding questions were created for the focus group meeting. Questions designed in a format to explore multiple things

  - Opinion of experts about solution in terms of solving the main issue
  - Strengths and weaknesses of the proposed Solution
  - Technical, organizational and personal barriers towards following the solution
  - Suggestions for improvement

  Detailed guide is shown in appendix [E]

- **Focus Group agenda planner**

  A complete agenda for the session with event details and time duration was planned and shared with the participants of the session. Details are shown in appendix[D]

- **Participants**

  Table 6.1 shows the details of the participants in focus group

Expert	Experience	Position/Title
Expert A	10+years	Product Owner
Expert B	10+years	Executive Member of Open Islamabad, Founding Member of Pakistan Agile Development Society
Expert C	7 years	Business Analyst
Expert D	8 years	Project Manager
Expert E	7 years	Assistant project Manager
Expert F	10 +years	Principal Software engineer/Scrum Master
Expert G	7 + years	PM /Scrum Master

Table 6.1: FG Participants details.

6.3. Expert opinion and analysis:

6.3.1. Data Analysis and results.

Audio/video taped and hand notes were translated onto excel sheets for analysis. Following is the compilation of the expert's opinion gathered according to the designed questions. Discussion is divided according to the sections created in guiding questionnaire.

❖ Opinion about solution:

All of the experts agree to the point that solution is solving the major issues as it is ensuring the inclusion of NFRs at all levels of SDLC. According to expert G and agreed by all the inclusion of NFRs at early phases is improving the whole process. Expert F further added that bringing all stakeholders at one page from the beginning of the process (about NFRs development) is making the later phases easier.

❖ Strengths of the solution

Experts of the session mentioned following strengths of the solution.

- Solution is Increasing overall product quality
- Solution has proposed NFRs management in all phases of SDLC
- Use of quality checklist is making the process easy to manage. (as no prolonged documentation would be required)
- The process will help in Reducing Cost during Maintenance
- Lower cost of re fixing the non-functional attributes.
- Solution is easy to understand and grasp.

❖ Barriers to solution adoption:

a) Technical barriers

For technical barriers experts thinks that their might not be any major technical barriers as most of the process is to be undertaken by the PO. If teams get full awareness about the outcomes of the product i.e. increased quality there will be no barrier in solution adoption , otherwise the solution is simple to follow and easy to understand.

b) Organizational barriers

Experts in the session mentioned the problem of mindset followed in organization. According to them, organizations are always reluctant to change, however proper cost benefits should overcome any organizational barriers. They further added that mindset of people is a problem but will change gradually if they carefully analyze the outcomes of following the proposed solution, and its long term benefits how it helps in decreasing technical debt and increment in Return of Investment in long term.

c) Personal barriers

For personal barriers experts mentioned that their might be some problems like lack of understanding of solution (at personal level) or inadequate knowledge and passion of a person. Likewise reluctant attitude towards adoption of new things, but all these barriers are easily manageable.

6.3.2. Expert suggestions for improvement

Following are the improvements suggested by the experts.

1. Checklist (quality matrix created for identification) should be concrete (explanation: assign values or numbers to the attributes so we will put the number in the story card.)
2. Eliminate quality discussion from daily scrum. As focus will be diverted (agreed by majority)
3. Use of Prototypes mentioned in identification is not providing major help, As it is more helpful for collecting Functional requirements; so it can be eliminated but few experts suggested that as it helps in identifying few UI related requirements so we can take help of prototypes where needed.

6.4. Improvements in Solution:

In accordance to expert's suggestions and guidance, Improvements made in the solution are as follow:

The supporting checklist for identification of NFRs is redesigned by adding values in form of levels for every quality attribute. Table 6.2 shows the updated version of the checklist. First column is the quality attributes.

Second is the sub quality attribute category, and then in correspondence to every quality attribute a 5 columns for 5 levels are created. The scale of the levels is mentioned in table 6.3. Now according to the product's requirement PO can choose between level 1 to 5 for every attribute. This level assignment to every quality attribute not only is helping in exact identification of quality requirement but also will help creating a scorecard at the end of the projects or end of sprint. That will help us decide number of things for next iterations also for other projects. For example:

1. Best ways to achieve quality (attribute wise) at any particular level.

2. Identification of hurdles in achieving quality level wise.

3. Improvements can be suggested and planned accordingly.

Table 6.2. Improved version of checklist.

	Quality attributes	Sub quality attributes		Level 1	Level 2	Level 3	Level 4	Level 5
	Functionality	Suitability	□					
		Accuracy	□					
		Interoperability	□					
		Standards	□					
		Security	□					
	Reliability	Fault tolerance	□					
		Maturity	□					
		Recoverability	□					
	Usability	Understandability	□					
		Learnability	□					
		Operability	□					
	Efficiency	Time Based efficiency	□					
		Resource based efficiency	□					

	Maintainability	Analyzability	□					
		Changeability	□					
		Stability	□					
		Testability	□					
	Portability	Adaptability	□					
		Instalability	□					
		Conformance	□					
		Replaceability	□					

Table 6.3: Checklist Scale:

Levels	**Description**
Level 1	Very low
Level 2	Low
Level 3	Moderate
Level 4	High
Level 5	Very high

6.5. Summary of findings

Altogether, the experts appreciated solution. As for the identified major issues, they commented that they also have same issues in their organizations. The proposed solution is addressing all of them, which is the best part. According to them, the mentioned obstacles of solution adoption i.e. mindset of people towards adopting change can easily be handled once they experience the positive results.

Chapter 7
Summary and Conclusion

7. Conclusion

Scrum is one of the most popular software development model followed in industry. As it is very successful in delivering Functional requirements of the projects. Somehow, it has failed in delivering nonfunctional requirements to its complete level, as suggested by literature and found in the study conducted.

In order to overcome the issue, this study proposed a set of guidelines that highlight NFRs management. It suggests the inclusion of NFRs in all scrum phases. For the purpose, easy and concise set of checklist is ted for every phase along with their supporting guidelines that is making it easier to understand and follow.

7.2. Contribution

The work aims to contribute the following:

- A research on management of NFRs in industry.
- Highlights the challenges industry is facing in managing NFRs in scrum environment.
- Causes of the identified challenges.
- A set of guidelines to resolve the issues, with a goal to improve the product quality.
- Feedback of the agile experts in form of a focus group session, in order to validate the proposed set of guidelines,
- Improvements in the proposed solution after expert feedback.

7.3. Limitation

Due to the limited time and resources, the research was restricted to a number of interviews/ organizations. Since findings were in line with the challenges mentioned in the literature, it does not affect the correctness of the study.

Secondly, for validation of the proposed solution, industry implementation wasn't made a part of the scope. If the solution is used by some of the software houses, that might be very helpful. However, this might take a lot of time. Since it was a one person research with limited time and resources, all aspects of the solution validation weren't possible to cover and focus group was chosen.

7.4. Future work

In reference to the future work, experts from the focus group suggested that Automation of designed checklist can also be the part of future considerations as it will help in keeping the records in more efficient way.
Moreover, implementation of solution in the industry will be the next target.
as this will help in fulfilling the real purpose of the research.

7.5. Summary of work

Research was started with the systematic review of literature; the major area focused in the beginning of the study was challenges of combining Agile methodologies with RE practices. The study leads us to the problem of mismanagement of NFRs with in agile environment. Literature highlighted issue of negligence of NFRs, but lack the detailed issues and causes of the subject. For the purpose, a survey was conducted to find out the challenges industry is facing while dealing with NFRs within Scrum environment. The scope of study was local software houses of Pakistan; Data collected from survey was than analyzed thematically in order to find out the major issues, causes and consequences of the issues. In order to overcome the identified issues a set of guidelines were devised and were converted in a form of checklist to make it easy and implementable in the industry. For validation of the study well known agile experts were invited in a focus group meeting and improvements were made in light of their recommendations. The experts approved and appreciated the solution with minor changes. The improvements were added into the solution accordingly.

8. References

[1] Ghafoor, Fawad, Ibrar Ali Shah, and Nasir Rashid. "Issues in Adopting Agile Methodologies in Global and Local Software Development: A Systematic Literature Review Protocol with Preliminary Results." *International Journal of Computer Applications* 160.7 (2017).

[2] Paulk, Mark C. "Agile methodologies and process discipline." *Institute for Software Research* (2002): 3

[3] "What is Agile Software Development?". Agile Alliance. 8 June 2013. Retrieved 4 April 2015

[4] Schwaber, Ken, and Jeff Sutherland. "The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. Scrum. org, July 2011." (2011).

[5] Asghar, Aneesa Rida, et al. "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development." *work* 7.12 (2016).

[6] Tsui, Frank, Orlando Karam, and Barbara Bernal. *Essentials of software engineering*. Jones & Bartlett Learning, 2016.

[7] Singh, Pratima, and Anil Kumar Tripathi. "Exploring problems and solutions in estimating testing effort for non functional requirement." *International Journal of Computers & Technology* 3.2b (2012): 284-290.

[8] Maiti, Richard R., and Frank J. Mitropoulos. "Prioritizing Non-Functional Requirements in Agile Software Engineering." *Proceedings of the SouthEast Conference*. ACM, 2017.

[9] Arbain, Adila Firdaus Binti, Imran Ghani, and Wan Mohd Nasir Wan Kadir. "Agile non functional requiremnents (NFR) traceability metamodel." *Software Engineering Conference (MySEC), 2014 8th Malaysian*. IEEE, 2014.

[10] Sachdeva, Vaibhav, and Lawrence Chung. "Handling non-functional requirements for big data and IOT projects in Scrum." *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on*. IEEE, 2017.

[11] Maiti, Richard R., and Frank J. Mitropoulos. "Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development." *SoutheastCon 2015.* IEEE, 2015.

[12] Farid, Weam M. "The Normap methodology: Lightweight engineering of non-functional requirements for agile processes." *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific.* Vol. 1. IEEE, 2012.

[13] Inayat, Irum, et al. "A systematic terature review on agile requirements engineering practices and challenges." *Computers in human behavior* 51 (2015): 915-929.

[14] Eberlein, Armin, and J. C. S. P. Leite. "Agile requirements definition: A view from requirements engineering." *Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE'02).* 2002.

[15] Um, Taehoon, et al. "A quality attributes evaluation method for an agile approach." *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on.* IEEE, 2011.

[16] Maiden, Neil, and Bianca Hollis. "Creativity on a shoestring: concept generating in agile development." *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems.* ACM, 2013.

[17] Paetsch, Frauke, Armin Eberlein, and Frank Maurer. "Requirements engineering and agile software development." *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on.* IEEE, 2003.

[18] Sillitti, Alberto, and Giancarlo Succi. "14 Requirements Engineering for Agile Methods." *Engineering and Managing Software*

Requirements (2005): 309-326.

[19] Paetsch, Frauke, Armin Eberlein, and Frank Maurer. "Requirements engineering and agile software development." *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on.* IEEE, 2003.

[20] Haugset, Borge, and Tor Stalhane. "Automated acceptance testing as an agile requirements engineering practice." *System Science (HICSS), 2012 45th Hawaii International Conference on.* IEEE, 2012.

[21] Ang, Jac Ky, et al. "Requirement engineering techniques in developing expert systems." *Computers & Informatics (ISCI), 2011 IEEE Symposium on.* IEEE, 2011.

[22] Hochmüller, Elke. "The requirements engineer as a liaison officer in agile software development." *Proceedings of the 1st workshop on agile requirements engineering.* ACM, 2011.

[23] Kavitha, C. R., and Sunitha Mary Thomas. "Requirement gathering for small projects using agile methods." *IJCA Special Issue on Computational Science-New Dimensions & Perspectives* 3 (2011): 122-128.

[24] Patel, Chetankumar, and Muthu Ramachandran. "Story card based agile software development." *International Journal of Hybrid Information Technology* 2.2 (2009): 125-140.

[25] Farid, Weam M., and Frank J. Mitropoulos. "NORMATIC: A visual tool for modeling non-functional requirements in agile processes." *Southeastcon, 2012 Proceedings of IEEE.* IEEE, 2012.

[26] Farid, Weam M., and Frank J. Mitropoulos. "NORPLAN: Non-

functional requirements planning for agile processes." *Southeastcon, 2013 Proceedings of IEEE.* IEEE, 2013.

[27] Farid, Weam M., and Frank J. Mitropoulos. "Visualization and scheduling of non-functional requirements for agile processes." *Southeastcon, 2013 Proceedings of IEEE.* IEEE, 2013.

[28] Pooley, R. J., and A. A. L. Abdullatif. "Cpasa: continuous performance assessment of software architecture." *Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on.* IEEE, 2010.

[29] Jacob, Stacy A., and S. Paige Furgerson. "Writing interview protocols and conducting interviews: Tips for students new to the field of qualitative research." *The Qualitative Report* 17.42 (2012): 1-10.

[30] Spickard, Jim. "How to construct an interview protocol." Retrieved January 31 (2005): 2009.

[31] Gill, Asif Qumer, Deborah Bunker, and Philip Seltsikas. "Moving forward: emerging themes in financial services technologies' adoption." *Communications of the Association for Information Systems* (2015).

[32] Dybå, Tore, et al. "Qualitative research in software engineering." *Empirical Software Engineering* 16.4 (2011): 425-429.

[33] Williams, P.M., 2001. Techniques for root cause analysis. Proceedings (Baylor University. Medical Center), 14(2), p.154.

[34] Cao, Lan, and Balasubramaniam Ramesh. "Agile requirements engineering practices: An empirical study." *IEEE software* 25.1 (2008).

[35] Krueger, Richard A., and Mary Anne Casey. *Focus groups: A practical guide for applied research.* Sage publications, 2014.

[36] Jacob, Stacy A., and S. Paige Furgerson. "Writing interview protocols and conducting interviews: Tips for students new to the field of qualitative research." *The Qualitative Report* 17.42 (2012): 1-10.

[37] Kvale, Steinar. Doing interviews. Sage, 2008 (book)

[38] http://rawgraphs.io/ (tool)

[39] Dybå, Tore, et al. "Qualitative research in software engineering." *Empirical Software Engineering* 16.4 (2011): 425-429.

[40] http://www.bulsuk.com/2009/07/5-why-analysis-using-table.html (tool)

[41] Kontio, Jyrki, Laura Lehtola, and Johanna Bragge. "Using the focus group method in software engineering: obtaining practitioner and user experiences." *Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on*. IEEE, 2004.

[42] Bader, Gloria E., and Catherine A. Rossi. *Focus groups: A step-by-step guide*. Bader Group, 1998.

[43] https://en.wikipedia.org/wiki/ISO/IEC_9126 (model)

Acronyms and Abbreviations	
NFRs	Non- functional Requirements
SDLC	Software development lifecycle
PO	Product Owner
RCA	Root cause Analysis
DOD	Definition of Done

APPENDIX

Appendix –A: scrum events description

The scrum events.	
Scrum Definition	Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is: □ Lightweight □ Simple to understand □ Difficult to master Scrum is a process framework that has been used to manage complex product development since the early 1990s. Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve. The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage. The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout the body of this document.
The Sprint	The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints best have consistent durations throughout a development effort. A new

	Sprint starts immediately after the conclusion of the previous Sprint. Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective. During the Sprint: □ No changes are made that would endanger the Sprint Goal; □ Quality goals do not decrease; and, □ Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned. Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a definition of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product. Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cos
Sprint Planning	The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team. Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

	Sprint Planning answers the following: □ What can be delivered in the Increment resulting from the upcoming Sprint? □ How will the work needed to deliver the Increment be achieved? © 1991-2013 Ken Schwaber and Jeff Sutherland, All Rights Reserved Page | 9 Topic One: What can be done this Sprint? The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint. The input to this meeting is the Product Backlog, the latest product Increment, projected capacity of the Development Team during the Sprint, and past performance of the Development Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint. After the Development Team forecasts the Product Backlog items it will deliver in the Sprint, the Scrum Team crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.
Daily Scrum	The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is done by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one. The Daily Scrum is

	held at the same time and place each day to reduce complexity. During the meeting, the Development Team members explain: □ What did I do yesterday that helped the Development Team meet the Sprint Goal? □ What will I do today to help the Development Team meet the Sprint Goal? □ Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?
Sprint Review	Sprint Review A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration. This is a four-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box. The Sprint Review includes the following elements: □ Attendees include the Scrum Team and key stakeholders invited by the Product Owner; □ The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done"; □ The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved; □ The

	Development Team demonstrates the work that it has "Done" and answers questions about the Increment; □ The Product Owner discusses the Product Backlog as it stands. He or she projects likely completion dates based on progress to date (if needed); □ The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning; □ Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and, □ Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product.
Sprint retrospective	The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint. The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is a three-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process. The purpose of the Sprint Retrospective is to: □ Inspect how the last Sprint went with regards to people, relationships, process, and tools; □ Identify and order the major items that went well and potential improvements; and, □ Create a plan for implementing improvements to the way the Scrum Team does its work.
Product backlog	Product Backlog The Product Backlog is an ordered list of everything that might be needed in the product and is the

	single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering. A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, its Product Backlog also exists. The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value.
Definition of Done	Definition of "Done" When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment. The same definition guides the Development Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team's current definition of "Done."

Appendix B: Survey Questionnaire

General Questions		Answers
1.	Name of Interviewee?	
2.	Name of the organization?	
3.	For how long have you been using Scrum? Personal and organizational experience?	

S No	*Questions/Follow up Questions*	*Answers / observation*
1	Do you take projects within the company or you have customers outside the company as well?	
Follow up	If within the organization …who is takin requirements and from whom?	
2	How do you collect requirements from customers?	
3	Who (from the scrum team) collects the requirements?	
4	What is your criteria for collecting requirements? Do you categorize your requirements (questions) in any way?	
3	During requirement elicitation process, Do you include questions about NFRs?	

Follow up	IF No: At what stage do you identify your quality requirements of the system?	
4	Do you separately identify that these are NFRs and FRs or not?	
5	IS there any elicitation technique you follow for collecting NFRs?	
Follow up	Does it help you collecting all quality requirements of the system?	
Follow up	If no technique: How do you Identify NFRs of the system? [Example questions]	
7	While taking requirements from clients Q: Do they understand your questions easily… or the terms you use for NFRs specifically?	
Follow up	What do you do if clients are non-technical people and do not understand NFRs?	
Follow up	What are the challenges you face when clients are unaware of NFRs?	
8	What are the other issues you face during identification or collection of NFRs from clients?	

9	Is there any way to confirm that u have elicited all NFRs? (early at the time of requirement identification)	
10	Do u ever feel a need for having a technique for collecting quality requirements?	
Follow up	What are the reasons behind not using any technique for requirement elicitation?	
11	Do you validate or crosscheck NFRs with the clients?	
12	When (in which scrum event) do you perform analysis of requirements?	
13	What kinds of problems do you face during analysis?	
14	What kinds of changes are mostly found during analysis?	
Follow up	What do you if you find that NFRs are vague?	
Follow up	What is the reason behind vague requirements?	
15	Do u have any criteria or method for conforming your analysis?	
16	During the regular meetings with the clients (throughout the project) do they explicitly talk about quality requirements that they need with other functional requirements?	

17	How the requirements are briefed to the team members?	
18	In case if requirements are left than at what stage do you identify them? And what will you do?	
19	Do you ever feel that your customers are mostly focused and concerned about functionality and not about non-functional requirements?	
Follow up	Do you think it can be a cause for negligence of quality requirements?	
Follow up	If Yes: then what do you do?	
20	Once you have elicited your requirements, how will you write them down? (Expected answers: in the form of user stories? Or any other format? No writing them at all.)	
Follow up	Do you decide NFR s sprint wise?	
Follow up	If user stories Do you write a complete separate user story for a single NFR?	

Follow up	If not user stories? How are they added into the product backlog?	
Follow up	If not added into the pB how do u track the requirements (Sprint wise)? Or How do you decide that which requirements is needed when in which phase or iteration?	
21	Why NFRs are not written in the form of user stories? What are the problems?	
22	Do you face any problem while writing NFRs in the form of user stories?	
23	Do you ever face any kinds of problems because of lack of specification of NFRs?	
24	Do you discuss NFRs in every sprint?	
25	PB management NFRs stored in PB how are they prioritized? (criteria)	
Follow uo	**If developer's choice:** Do you equally consider functional and non-functional requirement?	
Follow up	**If according to Customers demand:** Do you ever feel that your customers are mostly focused and concerned about functionality and not about non-functional requirements?	

26	What are the major challenges of NFR prioritization ?	
27	If NFRs are not stored in PB? How do you manage to implement them? I.e. how do you decide the implementation time? (sprint wise)	
28	Do you use any software for managing product backlog? If Yes: Does it support managing constraints that are to be kept in PB throughout the project?	
Follow up	Does it support requirement interdependency	
29	How do you test NFRs?	
30	When NFR testing is performed? (in which phase)	
31	Do you test non-functional requirements along with your functional requirement?	
32	Do you follow any specific techniques for testing quality requirements? (Quality charts, standards, matrix , automated tools)	
Follow up	The techniques u use for testing fully assure testing of all quality requirements? (any particular issues)	
33	What are the challenges you face in NFR testing?	

34	What is your acceptance criteria of a project?	
34	Do you feel a need for specific technique for testing NFRs in order to avoid any kind of negligence?	
35	Do you test them at the end of each iteration along with others or you keep on validating them during all phases?	
37	What are the general causes for not focusing on NFRs seriously throughout the project?	
38	Would you like to share any failures or after affects you face because of mismanagement of NFRs	

Appendix C: process analysis thematic network

Sr. no.	Category code	Underlying concepts	Basic themes.
1	Client/client role	Mostly NFRs are not elicited from clients during initial elicitation meetings.	NFRs are not elicited initially.
		Sometimes Clients do not explicitly mentions NFRs during elicitation process.	NFR are not explicitly mentioned by clients.
		Clients are mostly focused about functional features.	Focus of clients on functional requirements initially.
		Clients assume that NFRs will automatically become part of the project.	Clients Assume automatic implementation of NFRs.
		Sometimes clients are unaware of NFRs. (non-technical clients)	Clients are unaware of NFRs.
		Few technical clients explicitly mentions NFRs initially that saves from missing NFRS.	Technical clients mentions NFRs explicitly.
		Sometimes clients mentions NFRs after receiving 1st or initial deliverable from the team.	Initial deliverables help clients identify NFRs.

2	PO and team role in elicitation	PO and team members are sometimes ignorant of NFRs.	PO and team are ignorant of NFRs.
		Focus of PO is on functional features in the beginning	Focus of PO is on functional requirements initially.
		In early phases NFRS are mostly assumed by PO and team members of the project.	PO and team assume NFRs mostly in the project.
		PO and team does not discuss NFRs explicitly initially.	PO and team does not discuss NFRs explicitly initially
3	Process of elicitation	In early elicitation meetings, generally NFRs are not discussed with the clients explicitly.	Lack of NFRs discussion early in the project.
		Lack of early discussion of NFRs leads to late identification of NFRs.	Late identification of NFRs.
		Initial deliverables help clients identify NFRs.	Identification of NFRs after initial deliverables.
		Explicitly mentioned NFRs are discussed early in the project.	Early identification of only explicitly mentioned NFRs.
		Requirement elicitation questionnaire lacks questions about NFRS.	Lack of NFRs questions in elicitation questionnaire.

		Organization's quality standard sometimes restrain early and explicit questions about NFRs	Reliance on organization's quality standard restrain from explicit elicitation of NFRs.
4	Artifacts in elicitation	Questionnaires designed for early elicitation meetings clearly lacks NFRs. Prototypes help in identification of NFRS. There is a preset/predefined list of NFRs in organization that are supposed to be implemented in every project	Lack of questions about NFRs in elicitation questionnaire. Prototypes help in identification of NFRS. Standard based identification of NFRs.
5	Client role in analysis	Clients provide their feedback on NFRS sometimes after initial deliverable provided by the team. In some cases clients give their feedback about NFRS at the end of the project.	Clients feedback on NFRs is not taken initially i.e during design and analysis Clients provide their feedback mostly after deliverables.
6	PO and team role in analysis	QA members of the team perform analysis on NFRs of the project.	QA members perform NFR analysis. PO and client interaction in case of confusion.

		In case of any confusion PO discusses problems with clients	
7	Process of analysis	Analysis of NFRs is performed by QA members of the team once for the whole project NFRs are not discussed in every planning meeting. NFR analysis after explicitly mentioning of NFRS In case of any confusion PO asks clients queries. Predefined list of quality requirements sometimes helps in analyzing requirements also sometimes restrain team from explicitly analyzing NFRs. Sometimes analysis of NFRs is performed at the time of testing project and clients are not involved in that process.	Analysis of NFRs is performed once for whole project. Lack of NFR discussion in planning meetings. Analyzing only explicitly mentioned NFRs PO and client interaction in case of confusion. Analyzing NFRs in accordance with predefined quality standard. Analyzing NFRs during testing No explicit method for analyzing NFRS.

		There is no explicit method for NFR analysis.	
8	Artifacts in analysis	Organization's set quality standard is used for the analysis of NFRS. Prototypes help in analysis of NFRs.	Standard based analysis Prototypes help in analyzing NFRs.
9	Client role in specification	Client specifically has no role in specification.	No involvement of client during specification.
10	PO and team role in specification	Some PO's and teams believe that specifying NFRs is a difficult task. Despite of specifying NFRS PO and team build NFRs on the basis of their prior knowledge, experience and assumptions.	PO and team find difficulty in writing NFRs. Reliance on prior knowledge of PO and developers instead of specifying all NFRs.
11	Process of specification.	NFRs are mostly not written in the form of user stories Specification of NFRs is considered a difficult task. Mostly Only explicitly identified NFRs are specified.	NFRs are not specified as separate user stories. Specification of NFRs is considered a difficult task. Only Explicitly mentioned NFRs are specified.

		As NFRs are identified mostly late in the project so NFRs remained unspecified. Sometimes test cases are used for the specification of NFRs. There is no explicit or defined method for specifying NFRs that leads to the missing requirements. Unspecified requirements are not added in to the product Backlog.	Late identification leads to lack of specification. Specification of NFRs in the form of test cases No explicit method for NFR specification. Unspecified NFRs are missed from product backlog.
12	Artifacts in specification	Separate user stories are not created for NFRs mostly NFRS are not part of Product backlog mostly. Test cases help in specification of NFRs.	No explicit user stories for NFRs mostly. NFRs are missed from product backlog. Specification of NFRs through test cases.
13	Client role in prioritization	Clients are mostly concerned about functional features of the project that automatically gives lower precedence to NFRS.	Focus of clients on FRs gives lower precedence to NFRS during prioritization.
14	PO and team role in prioritization	If not explicitly mentioned by clients, PO and team give higher priority to FRS.	PO and team focus on FRs while prioritizing

15	Process of prioritization	Unspecified NFRs cannot be added into product backlog. NFRs not being part of the product backlog are not prioritized. Focus of stakeholders on functional requirements leaves NFRS low priority requirements.	Unspecified NFRs cannot be added into product backlog. NFRs not part of product backlog are not prioritized. Focus of SH on functional requirements while prioritization, gives lower precedence to NFRs.
16	Artifacts in prioritization	NFRS are not treated as PB items,	NFRs are not part of PB.
17	PO and team role in Implementation	Team mostly implement NFRs directly based on their prior experience.	Reliance on PO and team's prior knowledge and experience for implementation.
18	Client role in testing	Feedback is taken from the client.	Client's feedback helps in testing.
19	PO and team role in Testing	QA people perform testing on the basis of their prior knowledge and experience.	QA personnel perform testing on the basis of prior knowledge and experience.

18	Process of testing	NFRs are tested in accordance with the set organization's criteria. Test cases are not created for all requirements. Unspecified requirements sometimes left untested. Check list are used for NFR testing lacks all NFRs.	Org. set Standard based testing Test cases are not created for all requirements Unspecified requirements sometimes left untested. Checklist used for NFR testing does not cover all NFRs,
19	Artifacts in testing	Organizations have quality standard that helps in testing of NFRs Test cases are created for testing the requirements.	Organizational Standard for testing Test cases are created for testing the requirements,

Basic themes	**Organizing themes**	**Global themes.**
NFRs are not elicited initially. {NFR are not explicitly mentioned by clients. Identification of NFRs after initial deliverables}.	NFR elicitation	Process based

Initial deliverables help clients identifying NFRs. Lack of NFRs discussion early in the project. Late identification of NFR (at the time of development) Early identification of only explicitly mentioned NFRs Lack of NFRs questions in elicitation questionnaire		
Focus of clients on functional requirements initially Focus of PO is on functional requirements initially PO and team does not discuss NFRs explicitly initially PO and team focus on FRs while prioritizing	Stakeholder – focus	Stakeholder-based
Clients are unaware of NFRs.	Stakeholder – awareness	Stakeholder-based

PO and team are ignorant of NFRs. Technical clients mentions NFRs explicitly.		
Clients Assume automatic implementation of NFRs PO and team assume NFRs mostly in the project.	Stakeholder – assumptions	Stakeholder-based
Clients feedback on NFRs is not taken initially i.e during design and analysis Clients provide their feedback mostly after deliverables PO and client interaction in case of confusion No involvement of client during specification.	Stakeholder-interaction	Stakeholder-based
Reliance on PO and team's prior knowledge and experience for implementation. QA personnel perform testing on the basis of prior knowledge and experience	Stakeholder-knowledge based	Stakeholder –based

Reliance on prior knowledge of PO developers instead of specifying all NFRs		
Prototypes help in identification of NFRS Prototypes help in analyzing NFRs.	technique – prototyping	Artifact – based
Organizational Standard for testing Standard based identification of NFRs. Standard based analysis	Organization quality standard-based	Artifact –based
Analysis of NFRs is performed once for whole project. Lack of NFR discussion in planning gs. Analyzing only explicitly mentioned NFRs Analyzing NFRs in accordance with predefined quality standard. Analyzing NFRs during testing.	NFR analysis	Process based

No explicit method for analyzing NFRS.		
NFRs are not specified as separate user stories. Specification of NFRs is considered a difficult task. Only Explicitly mentioned NFRs are specified. Late identification leads to lack of specification. Specification of NFRs in the form of test cases No explicit method for NFR specification. Unspecified NFRs are missed from product backlog.	NFR specification	Process based
No explicit user stories for NFRs mostly. NFRs are missed from product backlog	Product backlog – management	Artifact –based

Specification of NFRs through test cases Test cases are used for testing	Technique- test cases	Artifact –based
Unspecified NFRs cannot be added into product backlog. NFRs not part of product backlog are not prioritized. Focus of SH on functional requirements while prioritization, gives lower precedence to NFRs. NFRs are generally not part of PB.	NFR prioritization	Process based
Org. set Standard based testing Test cases are not created for all requirements Unspecified requirements sometimes left untested. Checklist used for NFR testing does not cover all NFRs,	NFR testing	Process –based

Appendix D: Issue analysis thematic network

Category Code	Underlying issues	Basic themes
Stakeholder based	Clients do not mention NFRs explicitly in the initial meetings	Lack of initial NFR identification by clients.
	Focus of the client is on functional features of the project initially. Focus of PO is on functional features of the product	Focus of stakeholders on Functional Requirements initially
	NFRs are mostly demanded by clients in review meetings/after deliverables.	Clients mention NFRs after reviewing deliverables.
	Clients are unaware of non-functional requirements.	Lack of awareness of clients about NFRs.
	Clients assume NFRs of the project to be built automatically.	Clients assumptions about automatic implementation of NFRs.
	PO does not discuss NFRs in initial meetings.	Lack of NFR discussion by PO in early requirement meetings.
	Reliance on developer's experience about handling of NFRs throughout SDLC.	Reliance on developer's experience for NFR management.
	Team and PO rely on standard for all the requirement management related activities. i.e identification of requirements, specification , validation etc.	Reliance on organizational quality standard for NFRS management

	Lack of NFRs questions in early elicitation questionnaire.	Lack of NFR related questions in requirement elicitation questionnaires
Process based	NFRs are not usually discussed early in the project.	Lack of early NFR discussion.
	NFRs are identified mostly after initial deliverables.	Identification of NFRs after deliverables.
	No separate methods for NFR elicitation	Lack of explicit methods for NFR elicitation.
	NFRs are sometimes left unidentified in the project.	NFR left unidentified throughout the project.
	NFRs are discussed once for whole project not in every sprint.	NFRs are not discussed in every sprint. (once for whole project)
	User stories are not written for NFRs mostly	No explicit user stories are created for NFRs
	Writing user stories for NFRs is sometimes difficult as they are not very clear in early phases.	User stories for NFRs are considered difficult to create.
	There is no proper format for specifying NFRS… ***sometimes NFRs from prototypes are also missed just because they are not defined properly.***	No explicit method for specification of NFRs.
	Prioritization of NFRs does not take place.	Lack of NFR prioritization

	(Resource allocation for NFRs is left out in the case where they are not specified)and not added in PB. Assignment of Business Values to the NFRs is missed.	Cost estimation and resource allocation for NFRs remain indeterminate.
	NFRS are not added into the updated records after change request.	Missing of NFRS from updated records after change requests.
	Test cases are not written for all NFRs.	Missing test cases for NFRs.
	Quality standard does not cover all NFRs … mostly usability and user interface check list are considered as standard of quality.	Organization's own Quality standard does not cover all NFRS.
	A single quality standard cannot work for all types of projects.	One quality standard for all types of project
	Sometimes all the testing of NFRs is based on the organizational quality standard that leads to incomplete testing of the requirements	Only standard based NFR testing
	Test cases are written only for explicitly mentioned NFRs. (Those that are built on the basis of developer's experience left untested	Test cases for only explicitly mentioned NFRs.
	No proper format for creating quality standard	No proper format for creating quality standard

Artifact based	Product backlog does not contain separate items for NFRs.	Product backlogs lacks NFR related items.
	Designed questionnaires (oral or written) for elicitation of requirements lacks questions about NFRs	Lack of questions of NFRs in requirement elicitation questionnaire
	Testing of NFRs is done according to the set criteria hence misses the requirements that are not present in there	Reliance on Standard for NFR testing

Appendix E: Focus Group Questionnaire

General questions

Name	
Company Name	
Designation	
Work experience	
Email ID	

Section I

1. Do you think that proposed solution is helping in solving the Major issues? (HOW)

2. What are the strengths and weaknesses of proposed solution?

Strengths	**Weaknesses**

Section II

3. Do you see any problem in solution comprehension?

Section III

4. In your opinion what are the technical barriers to solution adoption?

5. In your opinion what are the organizational barriers to solution adoption?

6. In your opinion what are the personal barriers to solution adoption?

Section IV

7. What aspects of solution can be improved and How?

Appendix F- Focus group agenda

Focus group agenda planner	
Estimated time	**Agenda item**
10-15 mins	**Introduction** • Self intro • Purpose of Focus Group… brief overview • Intro of Participants
20 -25 mins	**Warm Up** • Presentation of the topic
45-50 mins	**Feedback from Participants (Question Period and discussion)** • Writing down their thoughts about the solution • Individually expressing their views about solution • Combining the most common views • Discussion on the most common /similar views • Presenting the findings • Final individual say on the findings
15 -20 mins	**Summary and conclusion** • Summarize the key points. • Highlighting the most important points (based on priority)
	Thank you and refreshment